最大クリーク問題を用いた
複数等質テスト自動構成手法とその近似手法

石井 隆稔

電気通信大学大学院 情報システム学研究科

学位申請論文 博士 (工学)

2014 年 3 月

最大クリーク問題を用いた
複数等質テスト自動構成手法とその近似手法

博士論文審査委員会

主査:　植野 真臣　　教授
委員:　大須賀 昭彦　　教授
委員:　栗原 聡　　教授
委員:　田原 康之　　准教授
委員:　古賀 久志　　准教授

最大クリーク問題を用いた
複数等質テスト自動構成手法とその近似手法

石井 隆稔

要旨

本研究ではｅテスティングにおける複数等質テスト自動構成手法を提案・開発した．複数等質テストとは, それぞれのテストに含まれるテスト項目は異なるが, 統計的な性質 (例えば, 得点分布や項目反応理論に基づく情報量等) が等しいテスト群である．本手法の特徴は, 複数等質テスト構成を最大クリーク問題として解くことで, 与えられたアイテムバンク・テスト構成条件で最大数のテストを構成可能な点である．これにより従来手法より多くのテストを構成可能であり, よりアイテムバンクを有効活用可能である．しかし, 本手法の厳密な計算はコストが高く, 大規模なテスト構成では計算が困難である．そのために，さらに， 限られた計算量でテスト構成を行う乱数探索を用いた近似手法を提案した．これにより, 厳密法の指数時間計算量と多項式 空間計算量 を定数オーダーへと軽減できた．最後に提案手法の有効性を示すため, シミュレーション及び実データを用いた実験を行い, 他手法より多くのテストを構成できることを示した．

目次

図目次

表目次

アルゴリズム目次

第 1 章

緒言

テストは, 1. その結果が受験者に重要な影響を与えるハイステークス・テスト (資格試験や入試試験など) と, 2. その結果が受験者に重要な影響を与えないローステークス・テスト (学校で行われる小テスト, 診断テストなど) に大別できる. 一般に, ハイステークス・テストでは, 複数等質テストが必要になる場合が多い. 複数等質テストとは, 異なる項目により構成されているにもかかわらず, 各テストが等質であるようなテスト集合である. 既に現在,e テスティングの実施に関する標準規格 ISO/IEC 23988 [1] で, ハイステークス・テストでの複数等質テスト構成が条件として記載されている. 実際, 我が国で最大の国家試験である情報処理技術者試験でも e テスティングにおける複数等質テスト構成が実施されている [2].

これまで複数等質テストはテスト管理者の勘と経験により構成されてきた. しかし, 近年,e テスティングの普及に伴い, テストの自動構成が可能となりつつある [2–4].

e テスティングではアイテムバンクという項目データベース (項目内容, 項目ごとの出題領域や統計データなどを格納している) を用いる. これらの情報を利用し, ユーザ所望の性質を持つ項目の組み合わせ (つまりテスト) を計算機により自動的に出力することがテストの自動構成である. そして, 与えられたアイテムバンクから互いに等質な大量のテストを同時に作り出すことが複数等質テスト自動構成である. 複数等質テスト構成の重要な課題は, このアイテムバンクを有効利用するために, なるべく効率的に多くのテストを構成しなければならないことである.

一般に, 新規項目作成がテスト構成作業で最もコストが高く, 一項目を実用化するためにも, データ収集を含めて数ヶ月を要する場合がある. このため, 複数等質テスト構成手法は与えられ

たアイテムバンクとテスト構成条件で最大数のテストを構成できることが望ましい．本論ではこの目的のため, 厳密に/漸近的に最大数のテストを構成可能な手法の開発を目的とする．

第 2 章では, 複数等質テスト構成の先行研究を紹介する．複数等質テスト構成は, 得点分布や回答所要時間などのユーザが求める条件 (以降、テスト構成条件と呼ぶ) に合う項目の組み合わせを探索する組み合わせ最適化問題として扱われる．この条件をテスト構成条件と呼び, 近年は項目反応理論に基づくテスト情報量を構成条件に採用する研究が多い．本章では, まずこの項目反応理論を紹介し, 続いて 4 つの複数等質テスト構成手法を紹介する．

第 3 章では, 与えられた条件・アイテムバンク中から最大数の複数等質テストを構成可能な手法を提案する．多くの従来手法では, 与えられたアイテムバンクから最大数のテストを構成する保証はない．本提案手法はテスト構成を, グラフ理論の最適化問題である最大クリーク問題として解くことでこの保証を可能とする．具体的には, 構成条件を満たすテストを頂点とし, 等質なテスト間に辺を引いたグラフ構造からの最大クリーク探索としてテスト構成を行う．クリークとは, 任意の 2 頂点が連結している構造であり, このグラフ中のクリーク構造は等質なテスト群である．従って, このグラフ中で最大数の要素を持つクリークを抽出することで, 最大数の複数等質テストを構成可能である．ただし, 本手法は計算量が多く, 現実的な規模での使用には計算の打ち切りが必要となる．その場合でも, 本手法は従来手法より多くのテストを構成可能である．本章ではこれらを示すため, シミュレーションおよび実データを用いた実験を行った．

第 4 章では, 第 3 章で提案した手法 (以降, 厳密法と呼ぶ) の近似手法を提案する．厳密法は計算コストが高く, 現実的な規模での使用には, 何らかの工夫が必要である．例えば, 計算の打ち切りにより時間的計算量の問題は解決可能である．しかし, 依然として空間的計算量の問題は解決されない．これは, クリーク構造の探索のため, 等質性を表すグラフを主記憶上に保持する必要があるためである．このグラフの頂点数は, アイテムバンク中の項目数 (以降, アイテムバンクサイズと呼ぶ) やテスト内の項目数に対し組み合わせ爆発的に増加する．そのため, このグラフ構造全域を計算機の主記憶上に保持することは困難である．本手法では, 厳密法中のグラフ全域からの最大クリーク探索を, 部分グラフからの最大クリーク探索の繰り返しへ近似しこの空間的計算量の問題を緩和する．本章では, この近似を行うことにより出力されるテスト数がどの程度減少するかを実験により示し, その場合でも従来手法よりも多くのテストが構成可能であることを, シミュレーションおよび実データを用いた実験により示す．

最後に第 5 章では, 本研究で得られた主な研究成果を総括し，本論文をまとめるとともに本研究の課題について述べる

第 2 章

複数等質テスト構成における先行研究

テストの自動構成とは, ユーザが所望する条件に合うよう, テストに出題する項目の組み合わせを計算機により選び出すことである [5–12]. また, 複数等質テスト構成とは与えられたアイテムバンクから互いに等質な複数のテストを同時に構成することであり, これまでに多くの先行研究がある [12–32].

近年の研究では, 世界標準のテスト理論である [2–4] 項目反応理論に基づくテスト情報量をテストの構成条件に採用する研究が多い (例えば [20, 23–32]). 本研究でもこれらに従い, このテスト構成条件にテスト情報量を採用する. 本章ではまず, この項目反応理論を紹介し, 続いて代表的な 4 つの複数等質テスト構成手法を紹介する.

2.1 項目反応理論

項目反応理論 (Item Response Theory; IRT) は, テスト理論の一つであり, 現在世界中で最も多用されているテスト理論である [2–4, 33, 34]. TOEFL や TOEIC, さらに日本でも医師国家試験 (厚生労働省), 情報処理技術者試験 (独立行政法人情報処理推進機構, 経済産業省), 日本留学試験 (独立行政法人日本学生支援機構) などの実際の大規模テストが IRT によって運用されている. また,IRT は, コンピュータを用いたテスト (Computer Based Testing; CBT) や, コンピュータ適応型テスト (Computer Adapted Testing; CAT) を運用するうえでも重要な背景理論となっている. 本節ではこの IRT について紹介する.

2.1.1 項目特性曲線

項目反応理論の特徴は, IRT モデルと呼ばれる確率モデルを用いて, 受験者の能力や項目の難易度等の特性を推定しようとする点である.

項目反応理論の中で最も有名である 2-パラメータ・ロジスティックモデル (2 Parameter Logistic Model; 2PL) では, ある問題の正答確率を受検者パラメータ (能力値 θ) と項目 i のパラメータ (識別力 a_i, 難易度 b_i) を用いて次のようにモデル化する.

$$P(x_i = 1|\theta) \equiv p_i(\theta) = \frac{1}{1 + exp(-Da_i(\theta - b_i))} \tag{2.1}$$

ただし, x_i は

$$x_i = \left\{ \begin{array}{c|l} 1 & \text{項目 } i \text{ に正答する} \\ 0 & \text{それ以外} \end{array} \right.$$

とする. また, 定数 D の値は一般に $D = 1.701$ を用いる. この式 (2.1) によって描かれる曲線を項目特性関数 (Item Characteristic Curve; ICC) と呼ぶ.

式 (2.1) において難易度パラメータ b_i は正答確率が 0.5 となる θ の値を表しており, この値が高いと, 正答するために必要な能力値 θ も高くなる. また, 識別力パラメータ a_i の値は $\theta = b_i$ 付近の曲線の傾きに比例しており, この値が大きいほど, 受験者の能力値 θ に対して正答率の増加が敏感になる. したがって, 識別力パラメータ a_i の高い問題ほど, 受験者の能力値 θ の値を高精度に識別可能となる.

項目反応理論ではこのモデルにより, 項目や受験者を能力値という 1 次元尺度上に布置できる. これらの項目特性や受験者の能力パラメータはテストデータから推定する. 具体的な推定方法は, 同時最尤推定法, 周辺最尤推定法, ベイズ推定法等を用いる [2].

2.1.2 情報量関数

情報量関数 (Information Function) とは, 各項目の受験者能力値 θ に対するフィッシャー情報量を表し, 能力値 θ の受験者に対し, どの程度誤差を少なく能力値を推定できるかを表す関数である. 2PL モデルでの項目情報量関数 (Item Information Function; IIF) はフィッシャー

情報量を前述の ICC について求めたもので, 項目 i の情報量関数は以下のように定義される.

$$I_i(\theta) = D^2 a_i^2 p_i(\theta)[1 - p_i(\theta)] \tag{2.2}$$

また, テスト全体の情報量 (Test Information Function; TIF) は, ある項目への反応が別の項目への反応に影響を与えないという局所独立の仮定の下, IIF を用い, 以下のように表される.

$$I(\theta) = D^2 \sum_{i \in Test}^{n} a_i^2 p_i(\theta)[1 - p_i(\theta)] \tag{2.3}$$

ただし,$Test$ は該当するテストに含まれる項目の集合である.

TIF を用いることでテスト全体が能力値 θ の尺度上のどのあたりを最も精度よく測定できるかを評価することが可能である. これらの情報量を用いて, テストや項目の品質に理論的根拠を与えることや, 目的に応じたテスト構成を行うことが可能となる.

近年の複数等質テスト構成は, 一般に, 先に紹介したテスト情報量を構成テスト間で等質化する [20, 23–32]. なぜならば, テスト情報量はそのテストにおける能力値 θ の推定精度を表し, これが等しいテストは同精度での能力値推定が可能なためである. 本研究でも, このテスト情報量を等質化することにより, 複数等質テストの構成を行う.

2.2 複数等質テスト構成の先行研究

2.2.1 複数等質テスト構成

e テスティングでは, アイテムバンクという項目データベース (項目内容, 項目ごとの出題領域や統計データなどを格納している) を用いる. 複数テスト構成とは, アイテムバンク中から, それぞれ等質なテスト (項目の組み合わせ) 群を計算機により探索することである (図 2.1) [12–32].

ただし, 同じ項目の組み合わせが出題されることは e テスティングの運用上好ましくない. そこで一般には, それぞれのテスト間の共通項目数に上限を定め (これを重複項目条件と呼ぶ) どの二つのテストもこれ以下の共通項目数となるよう等質テスト群を構成する

また, ここでのテストの等質性とは, 先に紹介した項目反応理論におけるテスト情報量について等質化することである (例えば, [24, 26, 30, 32]).

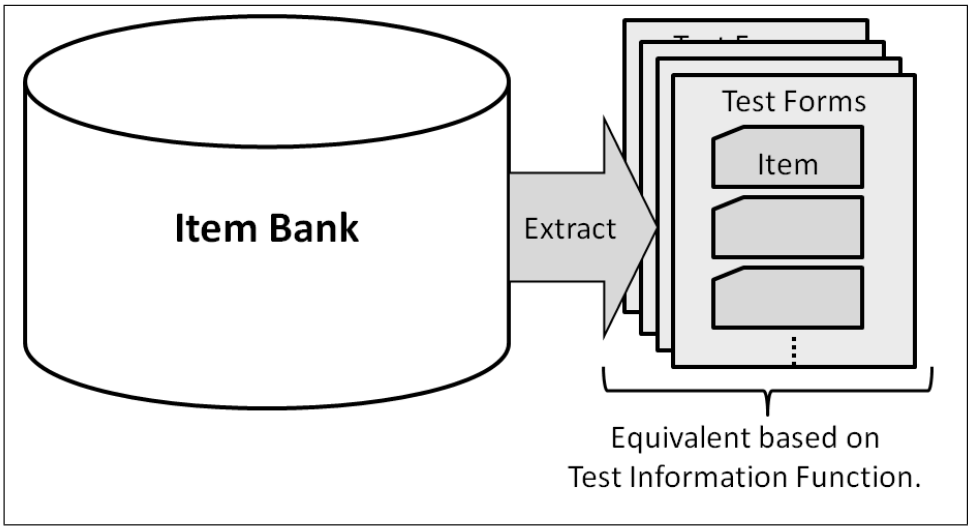

図 2.1　複数等質テスト構成の模式図

これらの研究は, どのような最適化問題 (以降テスト構成問題と呼ぶ) へと定式化を行うか, またそのテスト構成問題をどのようなアルゴリズムで解くかが異なっている. 本章では, いくつかの複数等質テスト構成手法を紹介していく.

2.2.2　領域別テスト構成

テストは内容的に分割された出題領域を持ち, それぞれの領域の出題項目数は明確に決まっている場合が多い. 例えば大学院での数学試験であれば, 微分積分, 線形代数, 解析数学, 確率・統計などのように出題領域がわかれている.

実際, 日本の国家試験である情報処理技術者試験「IT パスポート」も 79 領域に分かれており, それぞれ 1∼4 問, 合計で 100 問の出題数である. 合計で約 9000 項目を持つアイテムバンクはそれぞれの領域で分割されており, 各領域の平均項目数は 100∼200 台程度である.

また, リクルートキャリア社が提供する人事測定 e テスティングでは, 7 領域, それぞれ 4 題ずつ, 合計で 28 問の出題数である. 合計で約 1000 の項目を持つアイテムバンクはそれぞれ 80∼200 台程度の領域に分割されている.

このようなテストを構成する場合, 領域別のアイテムバンクから部分テストを独立に構成し, それらを統合してテスト構成を行う領域別テスト構成が一般に用いられる. これにより, 出題領

域の一様性を制御できるだけでなく, アイテムバンクの分割により計算量を大きく減らすことができる. このような工夫がこれまでの研究で使用されており, 本研究でもこの領域別テスト構成を前提としている.

2.2.3 線形計画法を用いた手法

van der Linden(2005) は線形計画法を用いてテスト構成を行う手法を提案した [24]. この手法はアイテムバンクから逐次的に等質なテストを構成していく手法である. テストに選び出す項目群とアイテムバンクに残す項目群を線形計画問題 (より正確には整数計画問題) として等質化し, テスト全体を等質化する.

具体的には, 所望のテスト数を R 個として, 以下の線形計画問題 R 回解く.

変数

$$
\begin{aligned}
& y && \geq 0 \\
& x_{ir} && = \begin{cases} 1 & i \text{ 番目の項目が} \\ & \text{現在構成中 } (r \text{ 番目}) \text{ の} \\ & \text{テストに含まれる} \\ 0 & \text{それ以外} \end{cases} \\
& z_i && = \begin{cases} 1 & i \text{ 番目の項目が} \\ & \text{シャドウテストに含まれる} \\ 0 & \text{それ以外} \end{cases} \\
& && (i = \{1, 2, \ldots, n\})
\end{aligned}
$$

minimize

$$y$$

subject to

$$\sum_{k=1}^{K} \sum_{i=1}^{n} |I_i(\theta_k) x_{ir} - \frac{1}{R-r} I_i(\theta_k) z_i| \leq y \tag{2.4}$$

(等質化のための条件)

$$\sum_{i=1}^{n} x_{ir} = g$$

$$\sum_{i=1}^{n} z_i = (R-r)g$$

(項目数条件)

$$\sum_{i=1}^{n} x_{ir} x_{it} \leq Overlap \quad (t = \{1, 2, \ldots, r-1\})$$

(重複項目数条件)

($x_{it}(t = \{1, 2, \ldots, r-1\})$ は定数. これ以前のテスト構成の結果を格納している)

(また, 領域別項目数等も制約条件に含まれる) .

ここでの,$I_i(\theta_k)$ は項目 i の θ 上のサンプル点 θ_k $(k = 1, \ldots, K)$ での項目情報量である. また,g はテストに含まれる項目数, $Overlap$ は最大で許される 2 テスト間の重複項目数である.

この線形計画問題は, θ_k 上での構成テストの情報量とアイテムバンクに残す項目群 (シャドウテストと呼ばれる) を変数 y を媒介とし等質化している. シャドウテストには今後構成する $R-r$ 個のテストの項目が含まれているため, この線形計画問題で等質化されている $\frac{1}{R-r} \sum_{i=1}^{n} I_i(\theta_k) z_i$ は, シャドウテスト中の 1 テストあたりの情報量平均である. また, 制約式

$\sum_{i=1}^{n} x_{ir}x_{it} \leq Overlap$ はこれまでに構成したテスト x_{it} と現在構成中のテスト x_{ir} との重複項目数の上限が $Overlap$ 以下となるように制限している.

この手法は, 後述するその他の手法と比べ変数の数が少なく最適な組み合わせ (最適解) を探す必要がある解空間が小さいため, 比較的低い計算量で複数等質テストを構成可能である. そのため, この手法はヨーロッパやアメリカで広く使われている.

しかし, 構成できたテスト数が与えられたアイテムバンクから最大数のテストを構成している保証はない. また, できるだけ多くのテストを構成したい, という使用方法が困難である. この手法は最初のテストを作る際に, いくつのテストを構成したいか (R) を入力する必要がある. できるだけ多くのテストを構成したい場合には, この値を徐々に増やしながら構成数が一番多くなる R を探す必要がある (Algorithm2.1).

アルゴリズム 2.1 Big Shadow Test method.

Require: Item bank and test constraints

Ensure: Uniform test forms

$R \leftarrow 0$.
$U_{max} := \phi$.
loop
  $R = R + 1$.
  Set $U := \phi$.
  $r = 0$.
  while $|U| < R$ **do**
    $r = r + 1$
    if Solving Problem (2.4) is success **then**
      The resulting test $\rightarrow \tilde{x}$ and add $\tilde{x}$ to the set U.
    else
      return U_{max}
    end if
  end while
  Set $U_{max} := U$
end loop

2.2.4 遺伝的アルゴリズムを用いた手法

Sun et. al. (2008) は遺伝的アルゴリズム (Genetic Algorithm) を用いてテスト構成問題を解く手法を提案した [30]. この手法は以下のテスト構成問題を遺伝的アルゴリズムを用いて解く.

変数

$$
\begin{aligned}
& y && \geq 0 \\
& x_{ir} && = \begin{cases} 1 & i \text{ 番目の項目が} \\ & r \text{ 番目のテストに含まれる} \\ 0 & \text{それ以外} \end{cases} \\
& && (i = \{1, 2, \ldots, n\}, r = \{1, 2, \ldots, R\})
\end{aligned}
$$

minimize

$$y$$

subject to

$$
\begin{aligned}
& \sum_{k=1}^{K} \sum_{i=1}^{n} |I_i(\theta_k) x_{ir} - T(\theta_k)| && \leq y \\
& && (r = \{1, 2, \ldots, R\}) \qquad (2.5) \\
& \text{(等質化のための条件)} \\
& \sum_{i=1}^{n} x_{ir} && = g \\
& && (r = \{1, 2, \ldots, R\}) \\
& \text{(項目数条件)} \\
& \sum_{i=1}^{n} x_{ir} x_{ir'} && \leq Overlap \\
& && (r < r', r = \{1, 2, \ldots, R\}, r' = \{1, 2, \ldots, R\}) \\
& \text{(重複項目数条件)} \\
& \text{(また, 領域別項目数等も制約条件に含まれる)}
\end{aligned}
$$

ここでの,$T(\theta)$ はユーザによって与えられる所望のテスト情報量である. すなわちこの最適化問題は,R 個全ての構成テスト情報量を目標となるテスト情報量 $T(\theta)$ に同時に近づける定式化である. そのため, 使用する変数の数は $nR+1$ となり, van der Linden 手法 [24] の $2n+1$ と比べ非常に大きい. そのため, 解空間も非常に大きくなり, 計算量も莫大になる. Sun らはこの問題を遺伝的アルゴリズムを用いて緩和した.

しかし,Linden [24] と同様に, 構成数が最大であることを保証しない. また, できるだけ多くのテストを構成したい, という使用方法にも Linden の手法と同様の工夫が必要である (アルゴリ

ズムが Algorithm2.2).

アルゴリズム 2.2 Genetic Algorithm method.

Require: Item bank and test constraints

Ensure: Uniform test forms

$R \leftarrow 0$.

Set $U := \phi$.

loop

$R = R + 1$.

if Solving Problem (2.5) is success **then**

Set the resulting tests to the set U.

else

return U.

end if

end loop

2.2.5 Bees Algorithm を用いた手法

Songmuang and Ueno (2011) は, Bees Algorithm を用いた複数等質テスト構成を提案している [32]. この手法は,(1) 条件を満たすテストを構成する,(2) 構成したテスト中より最も等質なテスト群を抽出する, という二つのステップから成り立つ.

具体的には, 以下のようなステップを行う.

Step A: (可能テスト構成)

以下の最適化問題を Bees Algorithm を用いて繰り返し解き, 条件を満たすテスト（“可能

テスト” と呼ぶ）を L 個構成する.

変数

$$
\begin{array}{lll}
y & \geq & 0 \\
x_{il} & = & \begin{cases} 1 & i \text{ 番目の項目が} \\ & \text{現在構成中 } (l \text{ 番目}) \text{ の} \\ & \text{テストに含まれる} \\ 0 & \text{それ以外} \end{cases} \\
 & & (i = \{1, 2, \ldots, n\})
\end{array}
$$

$$
\begin{array}{ll}
\textbf{minimize} & y \\
\textbf{subject to} & \\
\displaystyle\sum_{k=1}^{K}\sum_{i=1}^{n} |I_i(\theta_k)x_{il} - T(\theta_k)| & \leq \quad y, \\
\text{(等質化のための条件)} & \\
\displaystyle\sum_{i=1}^{n} x_{il} & = \quad g \\
\text{(項目数条件)} & \\
\text{(また, 領域別項目数等も制約条件に含まれる)} & .
\end{array}
\tag{2.6}
$$

Step B: (等質なテスト群の抽出)

StepB では StepA で構成した L 個のテスト中より, 最も等質なテスト群を抽出する. 具

体的には以下の最適化問題を Bees Algorithm を用いて解き, 複数等質テストを抽出する.

変数

$$s_l = \begin{cases} 1 & l \text{ 番目のテストが} \\ & \text{複数等質テスト群に含まれる} \\ 0 & \text{それ以外} \end{cases}$$
$$(l = \{1, 2, \ldots, L\})$$

minimize

$$\sqrt{\frac{1}{\sum_{l=1}^{L} s_l + 1} \sum_{l=1}^{L} s_l (e - \mu_S)^2} \tag{2.7}$$

(等質化のための条件)

subject to

$$\sum_{i=1}^{n} x_{il} x_{il'} \leq Overlap$$
$$(l < l', l = \{1, 2, \ldots, L\}, l' = \{2, 3, \ldots, L\})$$

(重複項目数条件)

ただし,

$$e = \sum_{k=1}^{K} |\sum_{i=1}^{n} I_i(\theta_k) x_i - T(\theta_k)| \tag{2.8}$$

$$\mu_S = \frac{1}{\sum_{l=1}^{L} s_l + 1} \sum_{l=1}^{L} s_l e \tag{2.9}$$

ここでの,e は fitting error と呼ばれ, それぞれの目標情報量からの二乗誤差を表している. また, 制約式 $\sum_{i=1}^{n} x_{il} x_{il'} \leq Overlap$ は複数等質テスト中の重複項目数の最大を制限している. 従って, この最適化問題では, どの二つのテストも重複条件を満たし fitting error の標準誤差が最小となる複数等質テストの組み合わせを出力する.

しかしこの手法も, 与えられたアイテムバンクから最大数のテストを構成する保証はない.

この手法の疑似コードは Algorithm2.3 である.

アルゴリズム 2.3 Bees Algorithm method.

Require: Item bank, test constraints, and the number of feasible test forms L

Ensure: Uniform test forms

(STEP A)

Set $U := \phi$.

repeat

Solve Problem (2.6) and add the resulting test to the set U.

until $|U| = L$

(STEP B)

return Solution of Problem (2.7).

2.2.6 集合充填問題を用いた手法

これまで紹介した手法は, 与えられたアイテムバンクから最大数のテストを構成している保証はない. 現実的には, アイテムバンクの構築は最もコストが高く, 与えられたアイテムバンクとテスト構成条件で最大数のテストを構成できるようにしたい.

これを実現する手法としてテスト構成数の最大化を行う手法がある. Belov and Armstrong (2006) は与えられたアイテムバンクから, 条件を満たす最大数の複数等質テストを構成する手法を提案している [26]. 彼らは複数等質テスト構成を集合充填問題（ Set Packing Problem）として解く. しかし, この手法は構成テスト間に項目重複を許さず, それぞれの項目は最大でも一度ずつしか出題できない. この理由から,Belov and Armstrong の手法は実用的には用いられていない.

Belov and Armstrong はテストを, それぞれ排他的な等質条件を満たす項目集合と定義し, アイテムバンクを最大数のテストへ分割する集合充填問題としてテスト構成を行う. すなわち,

以下のような最適化問題として定式化を行う.

$$
\begin{aligned}
&\textbf{変数} \\
&\qquad\qquad \mathbf{S}: \\
&\qquad\qquad \text{(与えられたアイテムバンクから構成可能なテストの集合)} \\
&\textbf{maximize} \\
&\qquad\qquad |\mathbf{S}| \\
&\textbf{subject to} \\
&\qquad \forall v, \forall w \in \mathbf{S} \quad , \quad v \cap w = \emptyset \text{(テスト間に共通項目を許さない)}. \\
&\qquad\quad v \in \mathbf{S} \qquad , \quad (v \text{ は与えられたテスト構成条件を満たす})
\end{aligned}
\tag{2.10}
$$

本手法はこれまで紹介した先行研究と異なり, テスト情報量誤差の最小化を行わない. 本手法では, テスト情報量は構成条件の一部として扱われる. 具体的には, テスト情報量の上限と下限を設定し, これを満たすものを等質と扱う. 例えば表 2.1 では $\theta = \{-2, -1, 0, 1, 2\}$ の点をサンプリングし, $\theta = -2$ の点には下限として $1.1 \leq I(\theta = -2)$ と上限として $I(\theta = -2) \leq 1.6$ を, $\theta = -1$ の点には下限として $1.5 \leq I(\theta = -1)$ と上限として $I(\theta = -1) \leq 2.0$ を, $\theta = 0$ の点には下限として $1.5 \leq I(\theta = 0)$ と上限として $I(\theta = 0) \leq 2.0$ を,... 設定している.

表 2.1 テスト情報量の上下限の例

Information Function (Lower Bound /Upper Bound)				
$\theta = -2.0$	$\theta = -1.0$	$\theta = 0.0$	$\theta = 1.0$	$\theta = 2.0$
1.1/1.6	1.5/2.0	1.5/2.0	1.5/2.0	1.1/1.6

この等質性条件を満たすよう, 最も多くの項目の集合 (テスト) へとアイテムバンクを分割するパターンを探索する最適化問題としてテスト構成問題を取り扱っている.

しかし, この手法は構成テスト間に項目重複を許さず, それぞれの項目は最大でも一度ずつしか出題できない. なぜならば, 最適化後のそれぞれの集合が排他的 (分割) であることが集合充填問題において定義されているためである. そのため, 本手法では (アイテムバンクサイズ)/(テスト項目数) がテスト構成数の最大となり, 十分なテスト数を構成できない. 例えば,1000 項目のアイテムバンクから 20 項目のテストを構成する場合, テスト間に項目の重複を許さない場合は最大で 50 テストしか構成することはできない. 一方, 先に紹介した従来手法では, テスト間に項

目の重複を許すことで構成テスト数を数倍から数十倍にまで増加させることができる．このような理由から,Belov and Armstrong の手法は実用的には用いられていない．

2.3 むすび

本章では, 項目反応理論と複数等質テスト構成の 4 つの先行研究を紹介した．これまで紹介したとおり, 実用されている手法では与えられたアイテムバンクから最大数のテストを構成している保証はない．また, 実用されていない手法ではあるが, 唯一,Belov and Armstrong(2006) は構成数の最大化を行っている．本手法はテスト間に重複を許さない条件でテスト構成数を最大化できる．

本論の主なアイデアは，アイテムバンクから最大数の複数等質テスト構成を可能にする Belov and Armstrong の手法を項目重複について拡張し, 従来手法より多くのテストを構成可能な効率の良い複数等質テスト構成を実現しようというものである．次章ではその具体的な方法について記述する．

第 3 章

最大クリーク問題を用いた複数等質テスト自動構成手法

3.1 はじめに

第 2 章では複数等質テスト構成の先行研究について紹介した. 実用されている手法は与えられたアイテムバンクから最大数のテストを構成している保証がないという問題があった. それが可能な手法としては Belov and Armstrong [26] の手法がある. しかし, この手法は構成テスト間に項目の重複を許さない条件でのみテスト構成可能であり, この条件がテスト構成数を著しく制限するため, 実用的には使用されていない. 一方, 実用されている手法では, テスト間に項目の重複を許すことで, 構成テスト数を数倍から数十倍にまで増加させることができる.

本論の主なアイデアは, 第 2 章で紹介した最大数の複数等質テスト構成を可能にする Belov and Armstrong [26] の手法を項目重複について拡張し, 従来手法より多くのテストを構成可能な効率の良い複数等質テスト構成を実現しようというものである.

具体的には, 集合充填問題の一般化である最大クリーク問題としてテスト構成問題を定式化する. これにより, 本提案手法はテスト間に項目重複を許す場合でも, 与えられたアイテムバンクとテスト構成条件で最大数のテストを構成可能である. しかし, 提案手法は計算コストが高く, 大規模なアイテムバンクでは計算を打ち切る必要がある.

本論文ではシミュレーション及び実データを用いた実験を行い, 計算を打ち切った場合でも,

本手法が他手法と比べ多くの等質テストを構成できることを示した.

3.2 提案手法

本論では,Belov and Armstrong の手法 [26] を重複項目について拡張し, 同一のアイテムバンク・構成条件で従来手法よりも多くのテストを構成する, 構成効率の良い複数等質テスト構成を提案する. 具体的には,Belov and Armstrong の手法 [26] で使用される集合重点問題の一般化である最大クリーク問題を用いてテスト構成を行う.

3.2.1 最大クリーク問題

最大クリーク問題とはグラフ理論の組み合わせ最適化問題の一つであり, 与えられたグラフから最大の頂点数を持つクリークと呼ばれる構造を探索する問題である. ここでクリークとは完全グラフ構造とも呼ばれる頂点の集合であり, 任意の 2 頂点が互いに連結されている構造を指す.

与えられたグラフを $\mathbf{G} = (V, E)$, ただし,V は頂点の集合, 辺の集合を E とし, 頂点 $v, w \in V$ が接続されているなら $\{v, w\} \in E$ としたとき, 最大クリーク問題は以下のように記述できる.

$$
\begin{aligned}
&\textbf{変数} && C \subseteq V \\
&\textbf{maximize} && |C| \\
&\textbf{subject to} && \forall v, \forall w \in C, \{v, w\} \in E.
\end{aligned}
\tag{3.1}
$$

この最大クリーク問題は集合充填問題の一般化となっている. V を与えられた集合のすべての部分集合, 辺 E を $v, w \in V, v \cap w = \emptyset \Rightarrow \{v, w\} \in E$ とした場合, 最大クリーク問題は集合充填問題となる [35].

3.2.2 複数等質テスト構成のための最大クリーク問題

本手法は, テスト構成問題を最大クリーク問題として取り扱う. 具体的には, 以下のグラフを考え, そこから最大クリークの探索・抽出を行い, 複数等質テストを構成する.

頂点: 与えられたアイテムバンクから構成可能なテストの等質条件を満たすテスト (以降,“可能テスト” と呼ぶ) 全てを頂点とする．ここでは Belov and Armstrong(2006) の手法 [26] と同様, テスト情報量の上限と下限が設定し, これを満たすものを等質であると扱う．

辺: 二つの可能テストが重複条件を満たす (重複項目数が重複条件 $Overlap$ 以下である) 場合その二つの頂点 (テスト) 間に辺を引く．

すなわち, 以下のような定式化を行う．

変数

$$C \subseteq V$$

maximize

$$|C|$$

subject to (3.2)

$$\forall v, \forall w \in C, \{v, w\} \in E.$$
$$\forall v \in V \quad , \quad (v \text{ はテスト構成条件を満たし等質})$$
$$\{v, w\} \in E \quad \Rightarrow \quad (v, w \text{ は重複条件を満たす,}$$
$$\text{つまり重複項目数が重複条件 } Overlap \text{ 以下である})$$

このグラフ中のクリークは複数等質テストである．なぜならば, このクリーク中の任意の 2 頂点は接続されており重複条件を満たしている．また, このクリーク中の頂点に対応するテストはそれぞれ等質である．従って, このグラフ中の最大クリークは最大の複数等質テスト群となる．このように, 複数等質テスト構成は最大クリーク問題として定式化でき, 理論的に最大数を保証した複数等質テストを出力する．

例えば, 図 3.1 は上のように構成したグラフである．図 3.1 中には 6 つの頂点 (テスト) と重複条件の満足を表す辺が 9 本ある．例えば,T5 と T6 はそれぞれテスト構成条件を満たす可能テストで, 重複条件を満たすため, 辺で結ばれている．このグラフ中の最大クリークは { T1, T2, T3, T4 } であり, これを抽出すると, 与えられたアイテムバンクから構成できる最大数の複数等質テストとなる．

この定式化は, Belov and Armstrong (2006) [26] の $\forall v, \forall w \in \mathbf{S}, v \cap w = \emptyset$(非重複条件) を $\forall v, \forall w \in C, \{v, w\} \in E$(重複条件を表す辺集合 E) に置き換えたものである．従って, 本手法は Belov and Armstrong (2006) [26] の重複条件について一般化したものである．

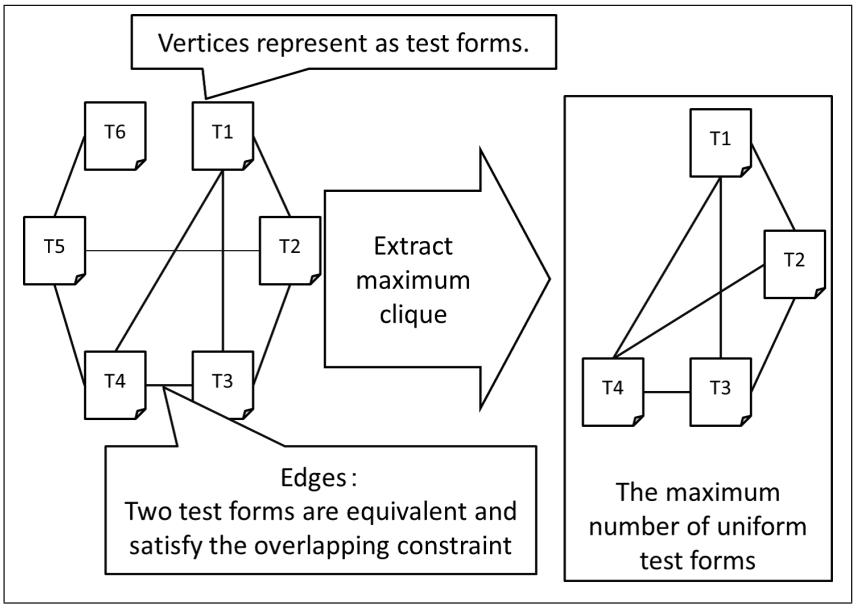

図 3.1　テスト構成のためのグラフ構造

3.2.3　アルゴリズム

提案手法の具体的なアルゴリズムを示す．本手法の疑似コードはアルゴリズム 3.1 のようになる．

アルゴリズム 3.1 最大クリーク問題を利用した複数等質テスト構成手法.

Require: Item bank and test constraints

Ensure: Uniform test forms

```
function MAIN
    (Step 1)
    V := φ.
    v := φ.
    items := given item bank.
    TESTGEN (v, items)
    (Step 2)
    E = φ
    for all v in V do
        for all u in V\v do
            if |v ∩ u| ≥ Overlap(テスト v, u の共通項目数が重複条件以下) then
                add {v, u} to E
            end if
        end for
    end for
    (Step 3)
    G := (V, E)
    中西, 富田のアルゴリズム [36] を使用して G から最大クリークを抽出する
return 求めた最大クリーク
end function
procedure TESTGEN(v, items)
    if |v| = g(与えられたテスト項目数) then
        if v がテスト構成条件を満たしている then
            add v to V
        end if
    else
        for all i in items do
            if v ∪ {i} のテスト情報量が与えられた情報量上限を下回っている then
                TESTGEN (v ∪ {i}, items\{i})
            end if
            items := items\{i}
        end for
    end if
end procedure
```

提案手法は以下の 3 つの Step からなる.

Step 1: (可能テスト構成)

疑似コード中の **(Step 1)** では与えられたアイテムバンクから構成条件を満たす, 可能テストを全て構成する. ただし, 全ての項目の組み合わせを探索するのではなく, 分枝限定法を用いて構成の効率化を行っている. 詳細は文献 [17] に譲るが, ここでは簡単に説明する.

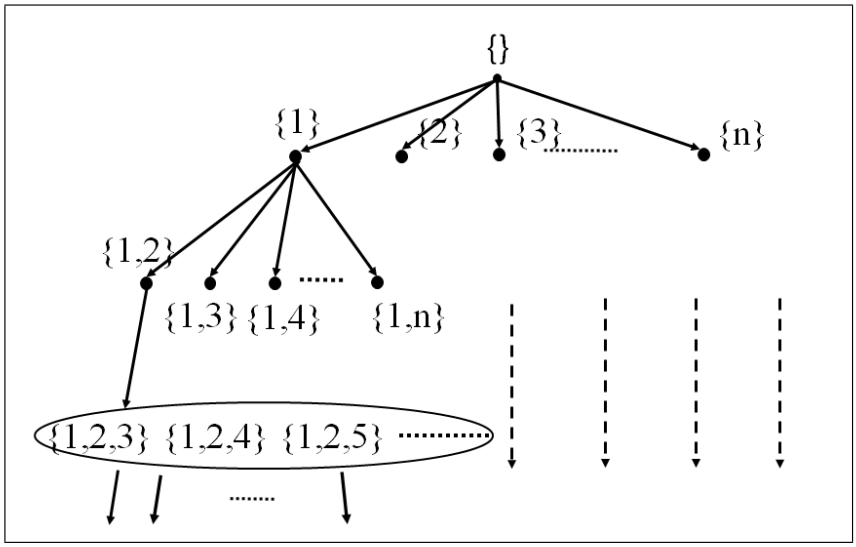

図 3.2　可能テスト構成のための探索木

図 3.2 は疑似コード中の手続 **TESTGEN**$(v, items)$ での可能テスト構成の探索木を表している. 図中の数字は項目を表し, それぞれのノードはテスト (項目の集合) を表している. 探索は, 深さ優先探索で行われ, 空集合 (根ノード) から一つづつ項目を追加し探索木を展開していく. この時, 各ノードをテストとみなし, 含まれている項目数が構成条件により指定された項目数以下であり, テスト情報量が構成条件により指定された上限を下回っている場合, 子ノードの展開を行う.

例えば, 図中では, まず空集合 (根ノード)“{}” を展開する. つまり, 項目を含んでいないテストにそれぞれの項目を追加し, テスト $\{1\}, \{2\}, \{3\}, \ldots, \{n\}$” を構成している.

次に,“{1}” のノードを展開する. つまり, 項目 2,3,4 がそれぞれ追加され, $\{1,2\}, \{1,3\}, \{1,4\}$ がそれぞれ構成される.

同様に,$\{1,2\}$ が展開され, $\{1,2,3\},\{1,2,4\},\{1,2,5\}$ が構成される. 仮に, 構成すべきテストの項目数が 3 であれば, これらのノードは展開されず, 項目数 4 以上のテストについては計算しない.

また,$\{1,2\}$ の展開が終了すると, $\{1,3\}$ の展開が行われる. このとき,$\{1,3\}$ のテスト情報量が与えられた上限を上回っている場合, $\{1,3\}$ は展開されない. なぜならば, 項目を追加してもテスト情報量は減少しないため, その先を展開しても情報量上限以下となることがないためである.

このように本手法では構成条件を使い条件を満たす全てのテストを列挙している.

Step 2: (テスト構成のためのグラフ生成)

(Step 2) ではテスト間の重複を表す, 関係グラフを構成する. Step 1 で構成した可能テストを頂点とみなし, その中の全ての 2 頂点について, 重複項目数を数え重複条件を満たすかを確認する. 重複条件を満たす頂点間に辺を引く.

Step 3: (最大クリークの抽出)

(Step 3) では Step 2 で構成した関係グラフから最大クリーク探索を行う. 本研究では, 現時点で, 最も高速であることが知られている中西, 富田のアルゴリズム [36] を用いて最大クリーク探索を行っている. 発見した最大クリークを複数等質テスト群として出力する.

計算量

また, この手法の時間的, 空間的計算量はそれぞれ,$\mathrm{O}(2^{0.19171F}),\mathrm{O}(F^2)$ となる. ただし,F は Step 1 で構成される可能テスト数である. 時間的計算量は各 Step で次のようになる. Step 1 では F 個の頂点の数え上げなので計算量 $\mathrm{O}(F)$, Step 2 では F 個中のすべてのペアを確認するので計算量は $\mathrm{O}(F^2)$ である. Step 3 では F 頂点からの中西, 富田のアルゴリズム [36] による最大クリーク探索を行たい, このアルゴリズムは頂点数 F に対し,$\mathrm{O}(2^{0.19171F})$ の時間計算量を持つ. そのため, この手法全体の時間的計算量は Step 3 に依存し $\mathrm{O}(2^{0.19171F})$ となる. 空間計算量は, 頂点数 F のグラフを保持する必要があるため,$\mathrm{O}(F^2)$ となっている.

3.3 評価実験

本提案手法は計算コストが高く, 現実的には計算の打ち切りが必要となる. 厳密に計算を行う場合と打ち切りを行う場合の本手法の有効性を示すため, 従来手法との比較実験を行った.

比較を行った従来手法は, van der Linden (2005) [24], Sun et. al. (2008) [30], Songmang and Ueno (2011) [32] である. Belov and Armstrong (2006) [26] はテスト間に項目重複を許さない条件での提案手法とアルゴリズム的にも同一となり, テスト構成数も本手法の結果と一致する. van der Linden (2008) [24] 中の線形計画問題解決には IBM 社の線形計画ソルバーである CPLEX [37] を用いた.

van der Linden (2008) [24] の手法は本提案手法で定義している等質性を満たさないテストを構成することがある. この手法は目標となる情報量に近いテストを逐次構成するが, 構成されるテストの情報量は, 徐々に目標情報量から離れていく性質を持つ. そのため, ある時点から本論中で定義した等質性 (情報量の上限下限) を満たさないテストを構成するようになる.

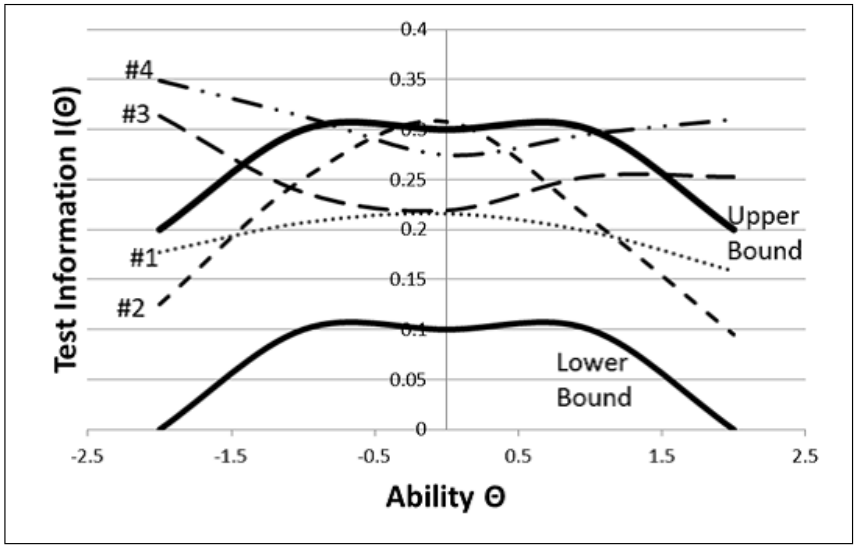

図 3.3 Linden 手法による構成テスト情報量

例えば, 図 3.3 は van der Linden (2008) [24] によって構成されたテストである. テストは # 1 ～ # 4 の順で構成されたが, # 3 以降のテストは, テスト情報量についての上限条件を満た

さないため, 等質性を満たさない. したがって, この場合, テストの構成は# 1,# 2 のみ成功したとみなし, 構成数を 2 とカウントしている.

実験では, 表 3.1 のような計算環境を用いた.

表 3.1 計算機環境

CPU	Intel(R) Core i7(R) 3930K 3.2GHz
System Memory	64.0GB
OS	Windows 7 SP1 64bit

多くの現実的な等質テスト構成では, 領域別テスト構成によりテスト構成を行う ([2–4, 26, 32] など). 本研究でも, このような領域別テスト構成を想定し実験を行った.

まず, 提案手法の厳密な振る舞いを調べるため, 小規模な 2 種類のアイテムバンク (サイズ $= 20, 30$) から 4 項目のテストを構成し, 提案手法と紹介した従来手法の構成数を比較した. それぞれの条件で 100 個ずつシミュレーションによりアイテムバンクを発生させ, その結果を比較した.

次に, 現実的な大きさのサイズの異なる 6 アイテムバンク (サイズ $= 70 \sim 120$) から 4 項目のテストを構成し, 計算の打ち切りを行った場合の提案手法と従来手法のテスト構成数を比較した. これらのシミュレーションは節 2.2.2 で述べた情報処理技術者試験やリクルートキャリア社の人事測定試験の条件を基に設定した.

3.3.1 厳密計算での構成数比較

この実験では, アイテムバンクサイズが違う 2 種のアイテムバンクをシミュレーションで発生させ実験を行った. それぞれのアイテムバンクサイズは $20, 30$ とした.

アイテムバンク中の項目は, 識別力パラメータ $a \sim U(0, 1)$, 困難度パラメータ $b \sim N(0, 1^2)$ として, データを発生させた. また, テスト構成条件は以下の通りである.

1. テスト項目数 $= 4$
2. 重複項目数の上限は $0, 1, 2$

3. 情報量条件は表 3.2 を与えた.

表 3.2　従来手法との比較 (小規模) のための情報量条件

Information Function (Lower Bound /Upper Bound)				
$\theta = -2.0$	$\theta = -1.0$	$\theta = 0.0$	$\theta = 1.0$	$\theta = 2.0$
0.0/0.2	0.1/0.3	0.1/0.3	0.1/0.3	0.0/0.2

比較を行った従来手法の目標情報量関数 $T(\theta_k)$ を情報量条件の上限下限の平均値として与えた.

これらの条件で, それぞれ計算を行った結果が, 表 3.3 である.

表 3.3　提案手法と従来手法のテスト構成数の平均・標準偏差比較

Item Bank Size	OC	BST		GA		BA		EM	
		Ave	SD	Ave	SD	Ave	SD	Ave	SD
20	0	0.89	0.63	0.76	0.65	0.93	0.67	0.97	0.72
	1	1.37	1.32	1.17	1.34	1.50	1.41	1.55	1.48
	2	2.31	1.98	2.39	3.21	3.34	3.72	3.65	4.43
30	0	1.51	0.72	1.25	0.78	1.64	0.93	1.81	0.86
	1	3.53	2.08	1.96	1.61	3.83	2.32	4.36	2.88
	2	5.56	2.27	3.77	3.55	9.78	6.17	13.06	10.60

表中の “OC” は重複項目条件を, “BST” は van der Linden (2005) [24] を, “GA” は Sun et. al. (2008) [30] を, “BA” は Songmuang and Ueno (2011) [32] を, “EM” は本章での提案手法を表している. また,“サイズ” がアイテムバンクサイズを表している.

また, 表 3.4 は,100 回中に提案手法が従来手法より多くのテストを構成した回数を示している.

ただし, “vsBST” が van der Linden (2005) [24] との比較結果を, “vsGA” は Sun et.al. (2008) [30] との比較結果を, “vsBA” は Songmuang and Ueno (2011) [32] との比較結果を示

表 3.4　提案手法が従来手法より多くのテストを構成した回数

Item Bank Size	OC	vsBST			vsGA			vsBA		
		>	=	<	>	=	<	>	=	<
20	0	0	92	8	0	79	21	0	96	4
	1	0	86	14	0	67	33	0	95	5
	2	0	68	32	0	43	57	0	85	15
30	0	0	70	30	0	48	52	0	84	16
	1	0	51	49	0	11	89	0	64	36
	2	0	27	73	0	5	95	0	35	65

している.

さらに, “>” は従来手法より提案手法のテスト構成数が少なかった場合の回数を, “=” は従来手法と提案手法のテスト構成数が同じであった場合の回数を, “<” は従来手法より提案手法のテスト構成数が多かった場合の回数を示している.

表 3.3 の結果より, 全ての条件で提案手法の平均テスト構成数が最も大きいことがわかる. また, 表 3.4 の結果より, 全ての条件で提案手法は従来手法以上のテスト数を構成できたことがわかる.

重複条件が 0 の場合の提案手法の結果は Belov and Armstrong [26] のものと一致するが, 他手法の重複条件が 1,2 の場合と比べ, テスト構成数が少ない結果となった. しかし, 重複条件が緩和されるにつれて提案手法は従来手法に比べ, 多くテストを構成できることがわかる. 加えて, この傾向は, アイテムバンクサイズが 20 の場合よりも,30 の場合のほうがより顕著である. これは, テスト構成の規模 (構成できるテスト数) が大きくなるほど, 提案手法は従来手法と比べ, アイテムバンクを有効活用できることを示唆している.

計算打ち切りによる近似度

次に, 計算の打ち切りを行った提案手法の振る舞いを検証する. 計算の打ち切りを行う場合, 構成テスト数が最大である保証はなくなる. また, 与える計算時間と出力される構成テスト数の

間にはトレードオフがあると考えられる．本節ではこのトレードオフ関係を明らかにするため, 計算開始からの経過時間と, その時点で出力される等質テスト数の関係をプロットした.

使用したアイテムバンク中の項目は識別力パラメータ $a \sim U(0,1)$, 困難度パラメータ $b \sim N(0,1^2)$ として発生させた．アイテムバンクサイズは $70, 80, 90, 100, 110, 120$ とした．また, テスト構成条件は以下の通りである.

1. テスト項目数 $= 4$
2. 重複項目数の上限は $0, 1, 2$
3. 情報量条件は表 3.5 の条件 1 を与えた.

表 3.5　領域別テスト構成のための情報量条件

Constraint ID	Information Function (Lower Bound /Upper Bound)				
	$\theta = -2.0$	$\theta = -1.0$	$\theta = 0.0$	$\theta = 1.0$	$\theta = 2.0$
1	0.1/0.2	0.2/0.3	0.4/0.5	0.2/0.3	0.1/0.2
2	0.0/0.2	0.1/0.3	0.3/0.5	0.1/0.3	0.0/0.2

この条件でテスト構成を行った計算時間と構成テスト数をプロットしたものが図 3.4,3.5,3.6 である.

図 3.4,3.5,3.6 の結果は, それぞれ, 重複条件が 0,1,2 の時の結果である．横軸は計算時間を, 縦軸は出力テスト構成数を示している.

図 3.4,3.5,3.6 いずれの結果も, 計算の開始から比較的短い時間で構成テスト数が収束している．つまり, 計算時間を長くしても出力されるテスト構成数はさほど増えないことがわかる．これは, 短い時間で計算の打ち切りを行っても比較的良い近似解を得られることの根拠になっている.

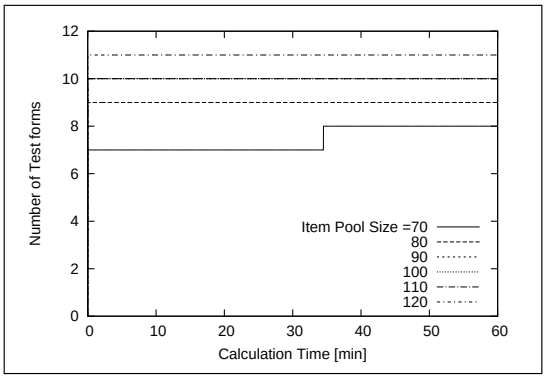

図 3.4　計算時間とテスト構成数 (非重複条件)

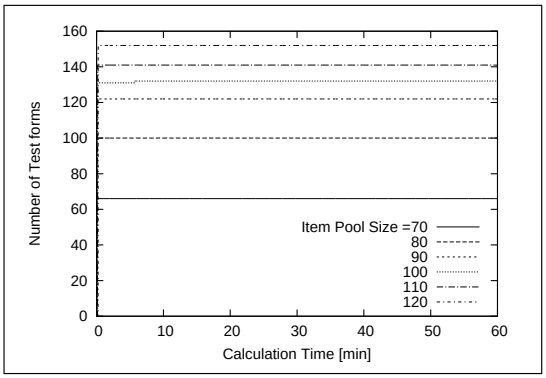

図 3.5　計算時間とテスト構成数 (重複項目数 1)

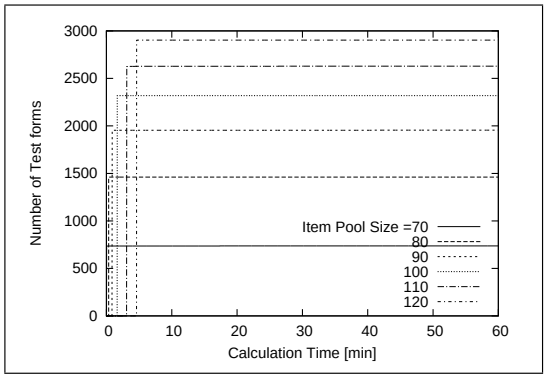

図 3.6　計算時間とテスト構成数 (重複項目数 2)

3.3.2　計算打ち切り時のテスト構成数比較

シミュレーションデータを用いた実験

最後に, 計算を打ち切った場合の提案手法のテスト構成数を従来手法のテスト構成数と比較した. 本実験でも, 節 3.3.1 で用いたアイテムバンク, 条件を使用した. 情報量条件は表 3.5 の条件 1,2 を与えた. 条件 2 に比べ条件 1 はテスト情報量の上下限の範囲が狭く, テスト構成数は少なくなる. また, それぞれの手法には計算時間の上限として 6 時間を与えた. これは, 節 3.3.1 の結果から 6 時間の計算時間は十分に出力テスト数を収束させる根拠となると考えたためである. 各条件でのテスト構成数を表 3.6 にまとめた. 表 3.6 中の † が付与された条件は 6 時間で計算終了しなかったもので, 記述はその時点で発見できた最大テスト数である. 今回の実験では, アイテ

表 3.6　提案手法打ち切り時の従来手法とのテスト構成数比較 (シミュレーションアイテムバンク)

Item Bank Size	OC	Constraint:1				Constraint:2			
		BST	GA	BA	EM	BST	GA	BA	EM
70	0	1	0	1	1	6	6	7	8†
	1	2	0	1	2	17	26	48	66†
	2	3	0	2	3	17	66	214	736†
80	0	2	1	2	2	7	8	8	9†
	1	11	2	11	12†	20	40	64	100†
	2	20	4	69	88†	20	82	242	1462†
90	0	2	1	2	2	8	7	8	10†
	1	13	3	11	13†	22	40	71	122†
	2	22	3	78	107†	22	81	251	1949†
100	0	2	1	2	2	8	7	8	10†
	1	13	3	11	12†	25	36	76	131†
	2	25	3	87	118†	25	80	292	2325†
110	0	2	1	2	2	8	8	9	10†
	1	13	3	11	13†	27	34	79	138†
	2	27	2	91	123†	27	70	308	2632†
120	0	2	2	2	2	9	6	9	11†
	1	13	2	10	13†	30	29	82	152†
	2	30	4	95	129†	30	68	336	2913†

† 6 時間で計算終了しなかったため, その時点で発見できている最大の等質テスト数を記した.

ムバンクサイズが大きな場合や, 等質性条件であるテスト情報量の上下限の範囲が広い条件で計算が終了しなかった.

表 3.6 より, 打ち切りを行った場合でも, 提案手法は従来手法と比較し数倍から数十倍も多くのテストを構成する結果となった. また, アイテムバンクサイズや重複項目条件を緩和すれば, さらに従来手法との構成数の差は大きくなった. 例えば, サイズ = 100, 条件 2, 重複条件 = 2 の場合, Songmang and Ueno の手法で 80 個のテストが構成できているが, 提案手法では 292 個構成

でき, サイズ = 120, 条件 2, 重複条件 = 2 の場合では Songmang and Ueno の手法で 336 個, 提案手法は 2913 個となり, 条件が緩和された時のテスト構成数の増加率を大きく改善できた.

従って, 提案手法は, 計算を打ち切った場合でも, 従来手法に比べ構成テスト数を増加させ, アイテムバンクを有効活用できることがわかる.

実データを用いた実験

次に, 実データを用いた実験を行った. 本実験では, リクルートキャリア社から提供された人事測定テストの項目データを用いて, 領域別テスト構成により, それぞれの手法のテスト構成数を比較した.

領域別テスト構成では, 領域別に構成したテストを統合したものが全体のテストとなる. そのため, 全領域を統合した統合テスト数は領域別の構成数以上とならない. (従って, それぞれの条件の領域別テスト構成数中で最も少なかったものが統合テスト数となる.)

使用したアイテムバンクは, リクルートキャリアが提供する人事測定 e テスティングのものである. 全体で約 442 の項目を持ち,4 つの領域に分かれている. それぞれの領域別の統計データは表 3.7 のとおりである.

表 3.7 実アイテムバンクの詳細

Item Bank Size	Parameter a			Parameter b		
	Range	Mean	SD	Range	Mean	SD
87	0.15∼0.67	0.35	0.134	-2.09∼4.55	0.73	1.625
93	0.19∼0.69	0.43	0.122	-3.92∼3.61	-0.79	1.196
104	0.13∼1.10	0.59	0.213	-0.18∼4.55	1.50	1.188
158	0.15∼3.08	0.44	0.255	-4.00∼4.00	-1.12	1.434
Total : 442	0.13∼3.08	0.46	0.217	-4.00∼4.55	0.01	1.806

この領域別アイテムバンクに対し, 以下の条件でテスト構成を行った.

1. 領域別テスト項目数 = 4
2. 重複項目数の上限は 0, 1, 2

3. 情報量条件は表 3.8 を与えた.

表 3.8　実データを用いた実験のためのテスト情報量条件

Information Function (Lower Bound /Upper Bound)				
$\theta = -2.0$	$\theta = -1.0$	$\theta = 0.0$	$\theta = 1.0$	$\theta = 2.0$
0.0/0.2	0.1/0.3	0.3/0.5	0.1/0.3	0.0/0.2

また, それぞれの手法には計算時間の上限として 6 時間を与えた.

表 3.9　打ち切りを行った提案手法と従来手法とのテスト構成数比較 (実アイテムバンク)

Item Bank Size	OC	BST	GA	BA	EM
87	0	3	3	4	4
	1	16	10	19	28
	2	21	36	139	304
93	0	4	5	5	6
	1	23	16	33	47
	2	23	43	211	660
104	0	6	5	8	10
	1	26	26	71	136
	2	26	59	275	2311
158	0	6	1	5	6
	1	22	12	24	37
	2	39	50	137	313
Total	0	3	1	4	4
442	1	16	10	19	28
	2	21	36	137	304

表 3.9 にそれぞれの領域, 全体での構成テスト数をまとめた. 提案手法の構成テスト数は, 従来手法を上回る結果となった. 従って, 実データに対しても, 提案手法はテスト構成数を増やし, アイテムバンクを有効活用できることがわかった.

3.4 むすび

本論文では,e テスティングにおいて, アイテムバンクを有効活用するためのテスト自動構成手法を提案した. 本手法はテスト構成を最大クリーク問題として扱うことにより, 与えられたアイテムバンクから条件を満たす複数等質テストを最大数構成可能である. ただし, 現実的には計算量の問題から計算の打ち切りを必要とする.

本手法の有効性を示すため, 本論文ではシミュレーション及び実データを用いた実験を行い, 計算を打ち切った場合でも, 本手法が他手法と比べ多くの等質テストを構成できることを示した. また, テスト構成の規模が大きくなれば, 提案手法は従来手法と比べ構成数を数倍から数十倍にまで増加させることも示した.

本手法の課題としては, より実用的な手法とするため, 計算量の軽減する方法を検討すべきである. また, 本論文中では, テスト間の重複項目数をどの程度許すのかについて議論を行っていない. この議論を行うためには, 一般にテストの目的や規模, 使用するアイテムバンクやテスト構成条件等, そしてどの程度の数のテストを構成したいのかを検討する必要がある. この議論についても今後の課題の一つとして, 取り組んでいきたい.

第 4 章

最大クリーク問題を用いた複数等質テスト自動構成近似手法

4.1 はじめに

第 3 章では, 与えられたアイテムバンク・テスト構成条件中で最大数の複数等質テストを構成する手法 (以降, これを厳密手法と呼ぶ) を提案した. しかし, この手法は計算量が非常に高いため, 現実場面での使用には計算の打ち切りや領域別のテスト構成等が必要となる. ただしその場合でも, 厳密手法の時間的・空間的計算量は指数的に増大するため, 例えば, 領域別のアイテムバンクが大きくなれば容易にメモリ不足により計算不能となる.

厳密法の時間的・空間的計算量はそれぞれ $\mathrm{O}(2^{0.19171F})$・$\mathrm{O}(F^2)$ であるここで F は可能テスト数であり, この F 自体もアイテムバンクサイズ n, テストの項目数 g に対して $F \propto {}_nC_g$ となっており, 組合せ爆発的に増加する. したがって, これらの計算量はアイテムバンクサイズに対し急激に増加する. そこで本章では厳密手法を近似化しこの問題を緩和を目指す. 具体的には, グラフ全域からの探索を行うのではなく, ランダムに取り出した部分グラフからの探索を繰り返す, 乱数探索アプローチ (例えば [38]) により, 計算量の緩和を目指す.

これにより本近似手法は, 厳密手法が計算を行うことが困難な場面でも従来手法より多くのテストを構成できる. また他の乱数探索手法 (例えば, [30,32]) と比較し, 解の探索空間をグラフ構造により限定するため, 効率よく多くのテストを構成できる.

これらの有効性を示すため, シミュレーションおよび実データを用いた実験を行い, 従来手法よりも多くのテストを構成できることを示す.

4.2 提案手法

4.2.1 厳密手法の問題点

厳密手法の時間的・空間的計算量は,F を可能テスト数として, それぞれ $\mathrm{O}(2^{0.19171F})$・$\mathrm{O}(F^2)$ である. この可能テスト数 F 自体も n をアイテムバンクサイズ,g をテストに含まれる項目数としたとき, 可能テスト数 F は $F \propto {}_nC_g$ の関係を持つため, これらの時間的・空間的計算コストは容易に実行可能な範囲を超える. 計算時間については計算の打ち切りを行うことで問題の緩和を行えるが, 空間計算量についての問題は依然残る. そこで本研究では, この空間計算量の問題を緩和するための近似手法を提案する.

既存の最大クリーク問題の近似解探索アルゴリズムはテスト構成には適用できなかった. なぜならば, これらのアルゴリズムはグラフ構造全体を主記憶上に保持していることを前提としているためである. 例えば, [39–41] 等はグラフ中の頂点次数を使用して探索を行うが, テスト構成のためのグラフ (関係グラフと以後呼ぶ) では頂点数 (= 可能テスト数 F) は組み合わせ爆発的に増加するため, グラフ構造全体を主記憶上に保持することはすぐに困難となる. そのため, 空間計算量を軽減するために近似手法はこのグラフ全体の構造を保持せずに最大クリークを探索できる必要がある.

4.2.2 アルゴリズム

そこで本近似手法は, グラフ全域からの探索を行う代わりにそのグラフの部分グラフからの探索を繰り返すことで, 最大複数等質テスト群の探索を行う. これにより, 部分グラフの構造のみをメモリ上に保持することとなり, グラフ構造全域を保持する場合と比較し空間計算量を軽減できる. この時, この部分グラフは全体グラフから抽出するのではなく, 直接生成する必要がある. なぜならば, 本手法中の全体グラフは規模が大きすぎるため保持が困難であり, これを生成してから部分グラフを選出することはできないからである.

また, 探索を行う部分グラフの特徴は探索の効率に影響を与える. たとえば, 最大クリークは

次数の高い頂点を含むことが多い．また, 次数の高い頂点の隣接頂点集合も最大クリークに含まれることが多い．したがって, そのような特徴を持つ部分グラフを高速に生成できれば, 探索の効率が上がる．しかし, このような部分グラフの生成は, 本手法では行っていない．なぜならば, 頂点次数についてはグラフ全体の構造が計算に必要であるため計算不能であった．また, 隣接頂点群の生成についても, 生成される隣接頂点群が多すぎて計算機上に保持できない問題や, 隣接頂点の生成自体に大きな時間を必要とし, 全体のパフォーマンスを落とすため採用しなかった．

したがって本手法では, ランダムに可能テストを構成し, そのグラフからの最大クリーク探索を行う．この繰り返しを与えられた計算時間中行い, その探索中で最も大きなクリークを出力する．これにより, 関係グラフ全域での最大クリーク (つまり, 最大複数等質テスト群) を漸近的に探索する．

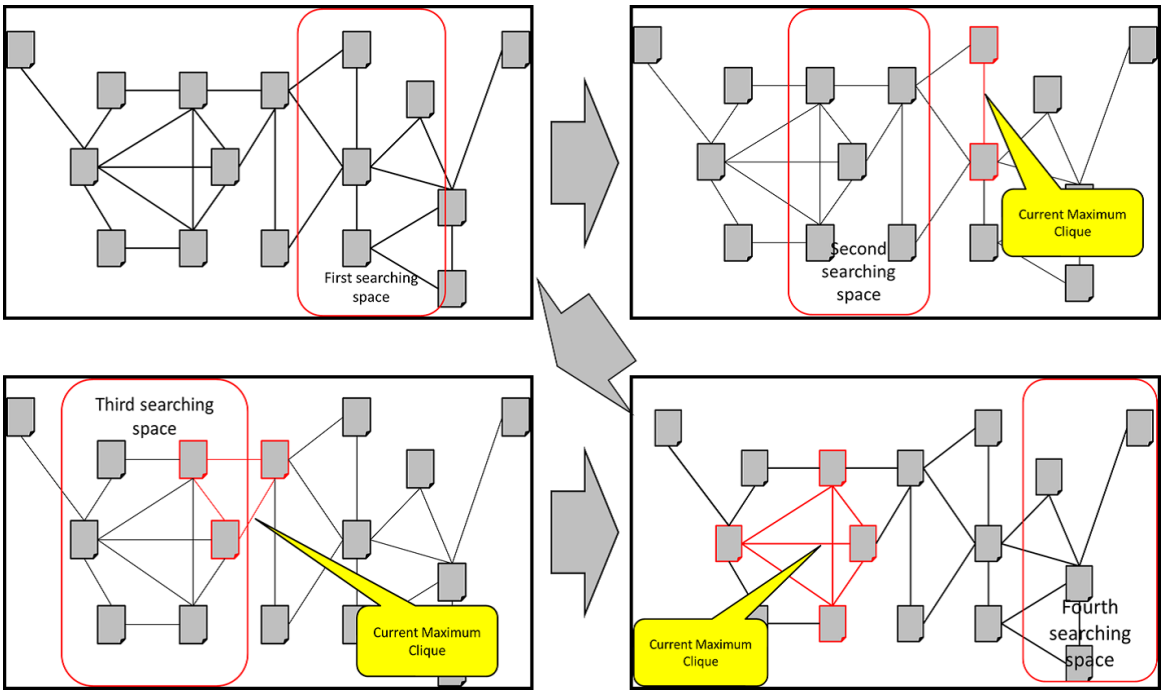

図 4.1　近似アルゴリズムの模式図

図 4.1 はこの近似探索の模式図である．図中のグラフは関係グラフ全域を表しているものとする．この例の探索は左上から始まっている．まず最初にランダムに部分グラフを選び出し (図中の赤で囲った部分), その中から最大クリークを探索する．最初の探索では部分グラフ中の最

大クリーク (サイズ=2) のものが見つかる．これは今まで見つかっている構造中で最も大きなクリークなので, 現在の最大として保持する (右上グラフ中, 黄色の吹き出しのグラフ構造)．同様に 2 度目の探索 (図中右上) では赤で囲った部分グラフからの探索を行い, サイズ=3 のクリークが発見され, 現在の最大クリークを更新する (左下グラフ中, 黄色の吹き出しのグラフ構造)．このような処理を繰り返し, 図中右下ではこの関係グラフ中で最大のサイズ=4 のクリーク (つまり, 複数等質テスト群) を発見している．

具体的なアルゴリズムを示す．本手法は以下の計算量条件を持つ．

C_1 一度の探索で使用する部分グラフの頂点数

C_2 一度の最大クリーク探索にかける時間

C_3 全体の計算時間

本アルゴリズムの疑似コードはアルゴリズム 4.1 のようになる．

アルゴリズム 4.1 近似手法.

Require: Item bank and test constraints
Ensure: Uniform test forms
function MAIN
$C_{max} = \phi$
(Step 1)
$V := \phi$.
$items :=$ given item bank.
repeat
$v :=$**TESTGEN2** $(v, items)$
add v to V
until $|V| <$ **C₁**
(Step 2)
$E = \phi$
for all v in V **do**
for all u in $V \backslash v$ **do**
if $|v \cap u| \geq Overlap$(テスト v, u の共通項目数が重複条件以下) **then**
add $\{v, u\}$ to E
end if
end for
end for
(Step 3)
$G := (V, E)$
中西, 富田のアルゴリズム [36] を使用して G から最大クリークを抽出し C とする. この時, 計算時間 **C₂** で計算を打ち切る
(Step 4)
if $|C| > |C_{max}|$ **then**
$C_{max} := C$
end if
if 累計の計算時間が **C₃** を超えていない **then**
GOTO (Step 1)
end if
return C_{max}
end function
function TESTGEN2$(v, items)$
if $|v|$ が与えられたテスト項目数 **then**
if v がテスト構成条件を満たしている **thenreturn** v
end if
else
repeat
chose i randomly from $items$.
until $v \cup \{i\}$ のテスト情報量が与えられた情報量上限を下回っている
return TESTGEN2 $(v \cup \{i\}, items \backslash \{i\})$
end if
end function

提案手法は以下の 4 つの Step からなる.

Step 1: (可能テスト構成)

(Step 1) では与えられたアイテムバンクから, ランダムに $\mathbf{C_1}$ 個の可能テストを構成する. 厳密法同様, 分枝限定操作を用いたテスト構成の高速化を本手法でも行っている. 厳密法で使用した可能テスト構成探索木 (例:図 3.2) をランダムにたどることでテスト構成を行う. 図 4.2 は疑似コード中の手続 **TESTGEN2**$(v, items)$ での可能テスト構成の探

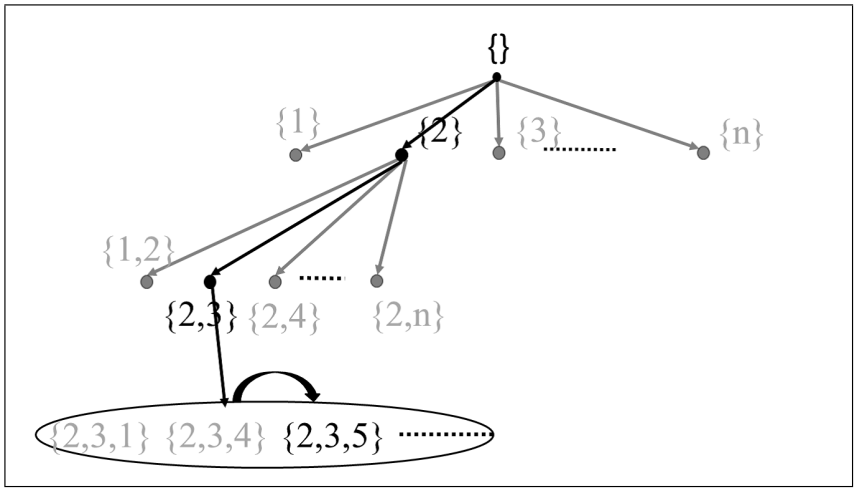

図 4.2 近似手法のための可能テスト探索木

索木を表している. 図中の数字は項目を表し, それぞれのノードはテスト (項目の集合) を表している. 探索は, 空集合 (根ノード) からアイテムバンクからランダムに選び出された項目を追加し探索木を展開していく. この時, 各ノードをテストとみなし, 含まれている項目数が構成条件により指定された項目数以下であり, テスト情報量が構成条件により指定された上限を下回っている場合, 子ノードの展開を行う.

例えば, 図中では, まず空集合 (根ノード)“{}” を展開する. つまり, 項目を含んでいないテストに項目を追加し, テストを構成する. ここでは項目 2 が選ばれたとして, テスト {2} を構成している.

次に,“{2}” のノードを展開する. ここでは 3 が選ばれたとして, $\{2, 3\}$ が構成される.

同様に,$\{2, 3\}$ が展開される. ただしこれらの手続き中で, 構成されるテストが情報量上限

を超えてしまう場合には, 追加を行わずに新たに別の項目を選びなおす. 仮に今, 項目 4 が選び出されたとする. ただし, テスト $\{2,3,4\}$ のテスト情報量が与えられた上限を上回っている場合, テスト $\{2,3,4\}$ は構成されず, 新たに追加する項目を選び直し (ここでは項目 5 が選ばれたとする), $\{2,3,5\}$ が構成される. このように構成条件を使い条件を満たすテストを列挙している. これを構成テストが $\mathbf{C_1}$ 個になるまで続ける.

Step 2: (テスト構成のためのグラフ生成)

厳密法での **Step 2** と同様に,**Step 1** で構成した全テスト間の重複項目数を数え, 重複条件を満たしていればそれらの間に辺を引き, テスト構成のための関係グラフを生成する.

Step 3: (最大クリーク探索)

厳密法での **Step 3** と同様に,**Step 2** で生成したグラフから最大クリークを探索する. ただし, この探索は $\mathbf{C_2}$ の計算時間で打ち切る.

Step 4: (終了条件判定)

Step 3 で発見された複数等質テストがこれまで見つかっている最大クリークよりも大きければ, それを現在見つかっている最大の複数等質テスト群として保持する. 全体での計算時間が $\mathbf{C_3}$ を超えていなければ,**Step 1** から再度探索を行う. 超えていれば, 発見できた最大の複数等質テスト群として現在見つかっている最大クリークを出力する.

4.2.3 計算量

本手法の時間的・空間的計算量はそれぞれ $\mathrm{O}(\mathbf{C_3})$,$\mathrm{O}({\mathbf{C_1}}^2)$ となる. 計算量が制御可能になったことで, 厳密法では困難であった条件でもテスト構成可能である. 可能テスト数 F が大きくなる場合, つまり, 大規模なアイテムバンクを与えられた場合やテスト情報量条件の上限と下限の範囲が広い場合に, 厳密法はメモリ不足により計算不能となる. しかし, 本近似手法では頂点数 $\mathbf{C_1}$ のグラフのみを保持するため, 空間計算量は $\mathrm{O}({\mathbf{C_1}}^2)$ となり, アイテムバンクサイズやテスト構成条件によらずテスト構成が可能である.

4.2.4 計算量条件と近似精度の評価

本手法には 3 つの計算量条件がある．これらのパラメータは出力するテスト数とのトレードオフがあると考えられる．本節では, この計算量条件と出力されるテスト数の関係を，様々な計算量条件においてテスト構成数を比較することで明らかにする．

ただし, 一度の最大クリーク探索にかける時間 $\mathbf{C_2}$ と構成テスト数の関係は, 厳密法での計算打ち切り時間と構成テスト数との関係と一致するため, 与える計算時間は比較的少なくても構成テスト数の大きな影響を与えないと考える (節 3.3.1)．そこで本節では,$\mathbf{C_1}, \mathbf{C_3}$ とテスト構成数との間の関係について検証する．

アイテムバンクサイズ $I = 120$ のシミュレーションアイテムバンクから, 以下のテスト構成条件を用いてテスト構成を行った．

1. テスト項目数 $= 4$
2. 重複項目数の上限は $0, 1, 2$
3. 情報量条件は表 4.1 を与えた．

表 4.1 計算量条件と構成数の関係を示すための実験用テスト情報量条件

Information Function (Lower Bound /Upper Bound)				
$\theta = -2.0$	$\theta = -1.0$	$\theta = 0.0$	$\theta = 1.0$	$\theta = 2.0$
0.1/0.2	0.2/0.3	0.3/0.4	0.2/0.4	0.1/0.2

計算量条件については, $\mathbf{C_1} = \{1000, 5000, 10000, 50000, 100000\}$, $\mathbf{C_2} = 60$ 秒, $\mathbf{C_3} \leq 1$ 時間として, テスト構成を行った．

結果が図 4.3, 4.4, 4.5 である．

図 4.3, 4.4, 4.5 はそれぞれ重複条件が $0, 1, 2$ の場合のテスト構成数を, 空間計算コスト条件 $\mathbf{C_1}$ ごとにまとめたものである．横軸に計算時間 $\mathbf{C_3}$, 縦軸にテスト構成数をプロットした．また, 図中の実線”Optimal”は厳密法で求めた最大のテスト構成数であり, これに近いほど良い近似に

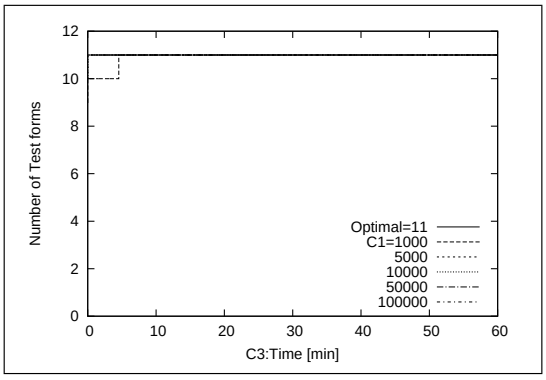

図 4.3　計算コスト条件 ($\mathbf{C_1}, \mathbf{C_3}$) と構成テスト数 (非重複条件)

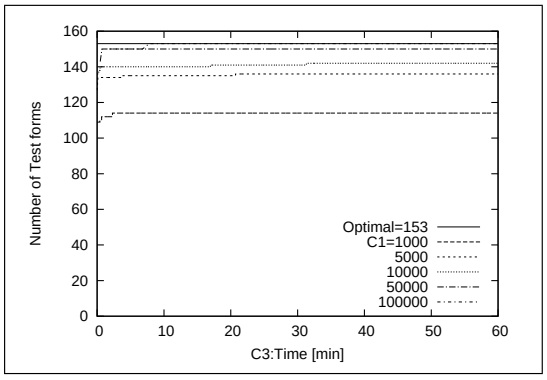

図 4.4　計算コスト条件 ($\mathbf{C_1}, \mathbf{C_3}$) と構成テスト数 (重複条件=1)

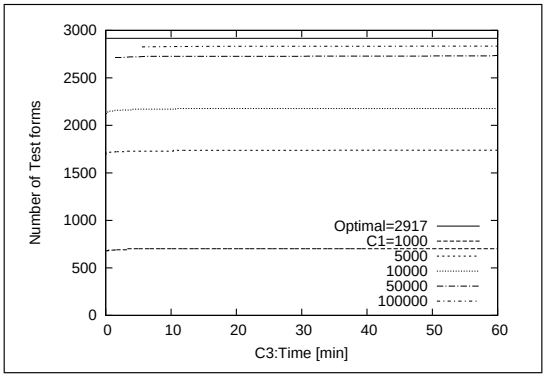

図 4.5　計算コスト条件 ($\mathbf{C_1}, \mathbf{C_3}$) と構成テスト数 (重複条件=2)

表 4.2　収束時でのテスト構成数.

	OC		
$\mathbf{C_1}$ Size	0	1	2
1000	11	115	705
5000	11	136	1740
10000	11	142	2181
50000	11	150	2738
100000	11	153	2839
Optimal	11	153	2917

なっている.

図 4.3, 4.4, 4.5 から計算時間 $\mathbf{C_3}$ を増やしてもあまり構成テスト数は増えないことがわかる. 重複項目数 0 の結果では, どの $\mathbf{C_1}$ 条件でも厳密手法と同じ最適解を発見できているが, 重複条件が増えた場合,$\mathbf{C_1}$ 条件が大きければ大きいほど良い近似となることがわかる.

そのため, これらの結果は以下のようにまとめられる.

1. 空間計算量条件 $\mathbf{C_1}$ は大きいほど, 本手法の近似精度は向上する. 計算環境の許す限り大きく与えることが望ましい.
2. 時間計算量条件 $\mathbf{C_2}$,$\mathbf{C_3}$ は共に, 出力されるテスト数にあまり大きな影響を与えないため,

短時間でも多くのテストを構成することが可能である.

4.2.5 厳密法との比較

本近似法の有効性を示すため厳密法との比較を行った. 厳密法と本近似法, そして紹介を行った従来手法によるシミュレーションアイテムバンクからのテスト構成数を比較した.

実験では, 識別力パラメータ $a \sim U(0,1)$, 困難度パラメータ $b \sim N(0,1^2)$ でシミュレーション項目を発生させ, サイズ $I = \{70, 80, 90, 100, 110, 120\}$ のアイテムバンクから以下の条件でテスト構成を行った.

1. テスト項目数 $= 4$
2. 重複項目数の上限は $0, 1, 2$
3. 情報量条件は表 4.3 を与えた.

表 4.3 厳密法と近似手法の比較実験用テスト情報量条件

ID	Information Function (Lower Bound /Upper Bound)				
	$\theta = -2.0$	$\theta = -1.0$	$\theta = 0$	$\theta = 1.0$	$\theta = 2.0$
1	0.1/0.2	0.2/0.3	0.4/0.5	0.2/0.3	0.1/0.2
2	0.0/0.2	0.1/0.3	0.5/0.3	0.1/0.3	0.0/0.2
3	0.0/0.4	0.1/0.5	0.7/0.3	0.1/0.5	0.0/0.4

本テスト情報量条件は ID:1<ID:2<ID:3 の順で上限と下限の範囲が拡大し, 構成テスト数が増えるよう設定した.

近似手法を除き, 計算時間は 6 時間を与えた. 近似手法へは $\mathbf{C_1} = 100000$(使用した計算環境が許す最大), $\mathbf{C_2} = 60$ 秒, $\mathbf{C_3} = 1400$ 秒, をそれぞれ与えた.

比較を行った従来手法の目標情報量関数 $T(\theta_k)$ は情報量条件の上下限の平均値を与えた. van der Linden (2008) [24] 中の線形計画問題解決には IBM 社の線形計画ソルバーである CPLEX [37] を用いた. 特に指定がない限り, 以降の実験でも従来手法はこの設定でテスト構成

を行った.

この結果をまとめたものが表 4.4 である.

表 4.4　厳密法と近似手法とのテスト構成数比較 (シミュレーションアイテムバンク)

Item Pool Size	OC	Constraint ID:1					Constraint ID:2					Constraint ID:3				
		BST	GA	BA	EM	RM	BST	GA	BA	EM	RM	BST	GA	BA	EM	RM
70	0	1	0	1	1	1	6	6	7	8†	7	7	7	7	8†	8
	1	2	0	1	2	2	17	26	48	66†	67	17	58	59	0‡	99
	2	3	0	2	3	3	17	66	214	736†	735	17	274	278	0‡	1767
80	0	2	1	2	2	2	7	8	8	9†	9	7	8	8	0‡	9
	1	11	2	11	12†	11	20	40	64	100†	100	20	74	78	0‡	131
	2	20	4	69	88†	88	20	82	242	1462†	1404	20	347	301	0‡	2825
90	0	2	1	2	2	2	8	7	8	10†	10	8	8	9	0‡	10
	1	13	3	11	13†	12	22	40	71	122†	119	22	83	86	0‡	156
	2	22	3	78	107†	107	22	81	251	1949†	1846	22	321	336	0‡	3634
100	0	2	1	2	2	2	8	7	8	10†	10	9	9	9	0‡	11
	1	13	3	11	12†	13	25	36	76	131†	130	25	88	87	0‡	173
	2	25	3	87	118†	118	25	80	292	2325†	2170	25	312	346	0‡	4288
110	0	2	1	2	2	2	8	8	9	10†	10	10	9	10	0‡	11
	1	13	3	11	13†	13	27	34	79	138†	137	27	86	92	0‡	195
	2	27	2	91	123†	123	27	70	308	2632†	2413	27	271	356	0‡	4938
120	0	2	2	2	2	2	9	6	9	11†	11	10	10	11	0‡	13
	1	13	2	10	13†	13	30	29	82	152†	150	30	92	102	0‡	229
	2	30	4	95	129†	127	30	68	336	2913†	2617	30	269	407	0‡	6006

†: 6 時間中で探索できた最大複数等質テスト数.

‡: メモリ不足により計算不可能.

表中の “BST” は van der Linden (2005) [24] を, “GA” は Sun et. al. (2008) [30] を, “BA” は Songmuang and Ueno (2011) [32] を, “EM” は厳密手法 “RM” は本章での提案近似手法を表している. また,“サイズ” がアイテムバンクサイズを表している.

情報量条件 ID:3 での多くの場合で厳密手法の計算が失敗 ($0^{\ddagger}$) していることがわかる. これは前述したとおり, テスト構成のための関係グラフがメモリ上に保持できなかったため計算不能となった. そのような条件でも本近似手法はテスト構成可能であり, 先行研究と比較して多くのテストを構成できることがわかる. また, 乱数探索を行う先行研究 (“GA”,“BA”) と比較して本

近似手法は, より短い時間で多くのテストを構成できており計算の効率が良いことがわかる. 加えて, テスト構成数が増えるほど, 従来手法との構成数差は広がることもわかる. ただし, 構成条件 ID:2 での結果を見ると, 構成数が増えるに従い厳密手法と近似手法の構成数差も広がり, 近似精度は悪くなることが示唆される.

したがって, 本実験での結果は以下のようにまとめることができる.

1. 厳密法は与えられたアイテムバンクテスト構成条件中で最大数であることが数学的に保障されたテスト群を構成可能であるが, 構成数が増えればメモリ不足により計算が困難となる.
2. 厳密法が計算困難なアイテムバンク・テスト構成条件であっても, 近似手法は計算が可能である. つまり, 計算コストの問題を緩和している.
3. 乱数探索を行う従来手法 ([30,32]) と比較し, より短い時間でより多くのテストを構成可能である. つまり, 等質テスト数を効率よく増加可能である.
4. 構成数が大きくなればなるほど, 従来手法との構成数差は広がり, 従来手法と比較しより有効にテスト構成が可能となる. ただし, 厳密手法との構成数差も広がり, 近似精度は下がることが示唆される.

4.3 評価実験

最後に本節では, 実際の使用を想定した条件で本近似手法と従来手法との比較を行った. まずはじめに, 領域別テスト構成を想定した条件で, 最後にそれを想定しない大規模なアイテムバンクからのテスト構成において, テスト構成数の比較を行った.

4.3.1 領域別テスト構成を想定したテスト構成数比較

シミュレーションデータを用いた比較

まず, 領域別テスト構成を想定し, 従来手法 [24, 30, 32] と本近似手法を比較した. 実験にはシミュレーションで発生させたアイテムバンクと実データを用いた.

シミュレーションデータは識別力パラメータ $a = 1$(1 パラメータモデルを仮定している), 困

難度パラメータ $b \sim N(0, 1^2)$ として発生させた. アイテムバンクサイズは $I = \{80, 100, 120\}$ として, それぞれ 100 のアイテムバンクを構成しテスト構成を行った. テスト構成条件は以下のものを使用した.

1. テスト項目数 $= 4$
2. 重複項目数の上限は $0, 1, 2$
3. 情報量条件は表 4.5 を与えた.

表 4.5　従来手法との比較のためのテスト情報量条件

ID	Information Function (Lower Bound /Upper Bound)				
	$\theta = -2.0$	$\theta = -1.0$	$\theta = 0$	$\theta = 1.0$	$\theta = 2.0$
1	0.7/1.5	0.8/1.6	0.8/1.6	0.8/1.6	0.7/1.5
2	0.1/0.9	0.2/1.0	0.2/1.0	0.2/1.0	0.1/1.0

近似手法の計算量条件は $\mathbf{C_1} = 100000$, $\mathbf{C_2} = 60$ 秒, $\mathbf{C_3} = 1$ 時間と設定した.

表 4.6 にそれぞれの条件でのテスト構成数の平均と標準偏差をまとめた. また, 表 4.7 はテスト構成数が従来手法を上回った回数である.

ただし, 表中の略号は第 3 章の節 3.3.1 と同様である.

表 4.6 の結果より, 全ての条件で提案手法の平均テスト構成数が最も大きいことがわかる.

表 4.7 の結果より, 全ての条件で提案手法は従来手法以上のテスト数を構成できたことがわかる.

これらは節 4.2.5 の結果のまとめ 3,4 を支持する結果となっている.

実データを用いた比較

本実験では, リクルートキャリア社から提供された人事測定テストの項目データを用いて, 領域別テスト構成により, それぞれの手法のテスト構成数を比較した.

使用したアイテムバンクは, リクルートキャリアが提供する人事測定 e テスティングのものである. 前章で使用した 4 アイテムバンクに加え, 厳密手法ではテスト構成できなかった 3 領域

表 4.6　近似手法と従来手法のテスト構成数の平均・標準偏差比較

Item Pool Size	OC	Constraint 1							
		BST		GA		BA		RM	
		Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD
80	0	7.07	1.60	7.96	1.73	8.93	1.60	11.15	2.49
	1	20.00	0.00	30.84	10.02	40.93	10.65	139.73	39.09
	2	20.00	0.00	52.63	17.97	76.48	18.25	2446.21	800.15
100	0	8.57	1.55	9.57	1.87	10.89	1.81	14.12	2.47
	1	25.00	0.00	35.00	10.96	55.04	12.28	218.32	52.40
	2	25.00	0.00	54.27	16.82	91.31	21.80	4696.10	1256.01
120	0	10.60	1.91	11.05	2.00	13.38	1.90	17.48	3.09
	1	30.00	0.00	40.41	11.67	70.94	15.88	318.26	66.74
	2	30.00	0.00	55.11	16.63	110.20	24.98	7933.48	1857.89
Item Pool Size	OC	Constraint 2							
		BST		GA		BA		RM	
		Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD
80	0	1.11	0.31	0.28	0.47	0.43	0.64	1.12	0.36
	1	2.32	1.74	0.34	0.61	0.96	1.77	2.34	1.77
	2	9.50	6.87	0.47	0.97	4.71	7.87	15.11	17.57
100	0	1.22	0.44	0.33	0.49	0.44	0.73	1.28	0.55
	1	2.93	2.21	0.43	0.87	1.12	2.19	3.19	3.10
	2	8.79	5.37	0.60	1.23	5.63	9.78	24.41	31.47
120	0	1.24	0.45	0.26	0.52	0.48	0.80	1.33	0.64
	1	3.28	2.39	0.39	0.87	1.47	2.87	4.07	4.41
	2	5.81	2.63	0.52	1.32	6.97	11.48	40.47	63.18

表 4.7　近似手法が従来手法より多くのテストを構成した回数

Item Pool Size	OC	Constraint 1									Constraint 2								
		vsBST			vsGA			vsBA			vsBST			vsGA			vsBA		
		>	=	<	>	=	<	>	=	<	>	=	<	>	=	<	>	=	<
80	0	0	0	100	0	1	99	0	9	91	0	99	1	0	21	79	0	33	67
	1	0	0	100	0	0	100	0	0	100	0	98	2	0	8	92	0	18	82
	2	0	0	100	0	0	100	0	0	100	0	54	46	0	0	100	0	1	99
100	0	0	0	100	0	0	100	0	2	98	0	94	6	0	16	84	0	22	78
	1	0	0	100	0	0	100	0	0	100	0	91	9	0	3	97	0	8	92
	2	0	0	100	0	0	100	0	0	100	0	38	62	0	0	100	0	0	100
120	0	0	0	100	0	0	100	0	1	99	0	91	9	0	7	93	0	22	78
	1	0	0	100	0	0	100	0	0	100	0	82	18	0	0	100	0	5	95
	2	0	0	100	0	0	100	0	0	100	0	20	80	0	1	99	0	0	100

を追加した合計 7 領域, 全体で 978 項目を持つアイテムバンクである．それぞれの領域別統計データは表 4.8 のとおりである．

表 4.8　実アイテムバンクの詳細

Item Bank Size	Parameter a			Parameter b		
	Range	Mean	SD	Range	Mean	SD
87	0.15–0.67	0.35	0.134	-2.09–4.55	0.73	1.625
93	0.19–0.69	0.43	0.122	-3.92–3.61	-0.79	1.196
104	0.13–1.10	0.59	0.213	-0.18–4.55	1.50	1.188
141	0.24–1.09	0.64	0.155	-1.41–3.91	0.60	0.855
158	0.15–3.08	0.44	0.255	-4.00–4.00	-1.12	1.434
175	0.12–0.93	0.39	0.139	-2.93–3.12	-0.25	1.113
220	0.16–0.92	0.46	0.155	-4.00–2.82	-1.28	1.098
Total : 978	0.12–3.08	0.46	0.198	-4.00–4.55	-0.22	1.572

この領域別アイテムバンクに対し, 以下の条件でテスト構成を行った.

1. テスト項目数 $= 4$
2. 重複項目数の上限は $0, 1, 2$
3. 情報量条件は表 4.3 を与えた.

また, それぞれの手法には計算時間の上限として 6 時間を与えた.

表 4.9 にそれぞれの条件・手法・アイテムバンクでのテスト構成数をまとめた.

表 4.9　近似手法と従来手法とのテスト構成数比較 (実アイテムバンク)

Item Pool	Overlap	Constraint 1				Constraint 2				Constraint 3			
Size	Constraint	BST	GA	BA	RM	BST	GA	BA	RM	BST	GA	BA	RM
87	0	0	0	0	0	3	3	4	4	3	3	4	4
	1	0	0	0	0	16	10	19	29	14	11	20	27
	2	0	0	0	0	21	36	139	307	21	39	140	309
93	0	0	0	0	0	4	5	5	6	5	5	5	6
	1	0	0	0	0	23	16	33	51	23	16	33	51
	2	0	0	0	0	23	43	211	658	23	54	208	721
104	0	2	2	2	2	6	5	8	10	12	15	15	18
	1	6	5	9	10	26	26	71	131	26	171	140	369
	2	26	14	83	121	26	59	275	2088	26	590	394	8442
141	0	10	3	9	10	18	19	21	27	26	31	27	35
	1	35	5	70	150	6	122	188	589	35	506	239	1014
	2	35	20	268	2307	10	185	393	11426	35	1511	386	19095
158	0	0	0	0	0	6	1	5	6	6	4	7	8
	1	0	0	0	0	22	12	24	40	39	42	75	131
	2	0	0	0	0	39	50	137	316	39	94	279	4877
175	0	2	0	2	2	6	6	7	9	6	6	8	10
	1	12	1	13	15	43	53	96	186	43	65	100	193
	2	43	2	128	234	43	102	303	7030	43	103	283	7413
220	0	2	0	2	2	7	5	8	10	9	8	10	13
	1	8	2	7	17	54	20	87	177	54	57	124	282
	2	54	8	75	136	54	44	309	5889	54	114	334	9938

いずれのアイテムバンク・テスト構成条件でも, 提案手法の構成テスト数は従来手法を上回る結果となった. 従って実データに対しても, 提案手法はテスト構成数を増やし, アイテムバンクを有効活用できることがわかった.

4.3.2 大規模アイテムバンクを想定したテスト構成数比較

最後に, 本近似手法の有効性を示すため, 領域別テスト構成を想定しない大規模アイテムバンクからのテスト構成数を従来手法と比較した.

実験には 3 つのシミュレーションアイテムバンクと 1 つの実アイテムバンクを用いた. シミュレーションデータは識別力パラメータ $a \sim U(0,1)$, 困難度パラメータ $b \sim N(0,1^2)$ として発生させた. アイテムバンクはそれぞれ $I = \{500, 1000, 2000\}$ の項目を持つ.

実アイテムバンクは, これまでの実験で使用したリクルートキャリア社から提供された人事測定テストのアイテムバンクを, 全領域合わせた 978 項目を一つのアイテムバンクとして扱い, テスト構成を行った.

これらのアイテムバンクから以下の構成条件でテスト構成した.

1. テスト項目数 = 25.
2. 重複項目数の上限は 0, 5, 10,
3. 情報量条件は表 4.10 を与えた

表 4.10 大規模テスト構成実験のための情報量条件.

Information Function (Lower Bound /Upper Bound)				
$\theta = -2.0$	$\theta = -1.0$	$\theta = 0$	$\theta = 1.0$	$\theta = 2.0$
1.0/2.0	2.0/3.0	2.0/3.0	2.0/3.0	1.0/2.0

この情報量条件はリクルートキャリア社の人事測定試験の条件を基に設定した.

近似手法の計算量条件は $\mathbf{C_1} = 100000$, $\mathbf{C_2} = 60$ 秒, $\mathbf{C_3} = 24$ 時間と設定した.

Table 4.11 が各手法・条件でのテスト構成をまとめたものである. 重複条件=0 の時を除い

表 4.11　大規模テスト構成における近似手法と従来手法とのテスト構成数比較

Item Pool Size	OC	Methods			
		BST	GA	BA	RM
500	0	12	3	5	10
	5	20	23	96	4380
	10	20	21	107	99983
1000	0	21	4	6	17
	5	40	17	104	46305
	10	40	19	105	100000
2000	0	53	8	12	32
	5	80	22	104	96876
	10	80	23	103	100000
978 (actual)	0	24	9	9	16
	5	39	283	371	40814
	10	39	286	381	100000

て本近似手法が最も多くのテストを構成できている.

重複条件=0 の場合は van der Linden (2008) [24] の手法が最も多くのテストを構成できている. また, 多くの重複条件=10 の場合は, テスト数が 100000 かそれに近い値に飽和している. これらの原因としては, 与えた計算量条件 $\mathbf{C_1}$ の値がテスト構成の規模に対して小さすぎることが考えられる. 構成テスト数が $\mathbf{C_1}$ に飽和している現象は, 一度の探索で使用する部分グラフの大きさよりも, 全域での最大クリークのほうが大きいことを示しており, これらの条件に対して十分な $\mathbf{C_1}$ を与えられていないことを示唆する. そのため,$\mathbf{C_1}$ に与える条件をさらに大きくすればテスト構成数は増えることが予想される. (ただし, 今回の実験で使用した $\mathbf{C_1}$ の大きさは, 使用計算機環境の与える最大であったため, これ以上大きな値は設定不能であった) しかしこのような条件でも, 提案手法のテスト構成数は重複条件を増やすことで従来手法以上に増加するため, 重複を許すことで, 本手法は従来手法以上のテストを構成可能である.

4.4 むすび

本章では, 前章で提案した厳密手法を, 乱数探索を用いて近似することで, 厳密手法での計算量の問題を緩和した. 具体的には, 関係グラフ全域からの探索をランダムに取り出した部分グラフからの探索の繰り返しに近似し, 空間的計算量を軽減した.

その結果, 本近似手法は, 厳密手法では計算困難である大規模なテスト構成についても従来手法よりも多くのテストを構成できることを示した. 一部の条件, つまり, 大規模なテスト構成における非重複条件でのテスト構成では, 従来手法よりもテスト構成数が少なくなってしまう場合もあるが, その場合でも, 構成テスト間に項目の重複を許すことで, 従来手法の数百倍程度のテスト数を構成可能なことを示した.

今後の課題としては, さらにさまざまな制約条件があるテスト構成にも対応したい. 例えば, 項目ごとの使用回数に偏りがあることは, アイテムバンク方式の e テスティングでは好ましくない. なぜならば, 多く出題される項目はインターネット等で暴露されやすく, 経年により急速に信頼性が失われるためである. 本手法では項目の出題回数は制御されていないため, 項目の出題回数に極端な偏りが生じ得る. そのため, それぞれの項目に対する制約条件を含んだ形での定式化ができることが望ましい. 今後はこのような点についても最適化可能な手法としていきたい.

第 5 章

結言

本論では,e テスティングにおける複数等質テスト構成手法の提案と評価について述べた.

第 2 章では複数等質テスト構成の先行研究について紹介を行った. 本章では代表的な 4 つの先行研究を紹介した. 実用されている手法は, 与えられたアイテムバンクから最大数のテストを構成する保証がない. また, それを保証する手法 (Belov and Armstrong [26]) はテスト間に項目の重複を許さない形の定式化であり, この条件はテスト構成数を著しく制限するため, 実用的には用いられていないことを述べた. 非重複条件は項目をそれぞれ一度きりしか使用しない条件である. 実用されている手法はテスト間の項目重複を許すことで, テスト構成数を数倍から数十倍にまで増やすことが可能であり, 実用化のためにはテスト間に項目の重複を許した条件でのテスト構成が必須であることを本章では述べた.

第 3 章ではテスト間に重複を許す条件において数学的に保証可能な最大数の複数等質テストを構成する手法を提案した. 本手法の主なアイデアは第 2 章で紹介した Belov and Armstrong [26] を重複項目条件について一般化することでこれを目指した. 本手法はテスト構成を, Belov and Armstrong [26] で使用されている集合充填問題の一般化である最大クリーク問題を用いて行う手法である. 具体的には, テストを頂点とし, 等質で重複項目数が一定数以下なテスト間に辺を引いたグラフからの最大クリーク問題として複数等質テスト構成をおこなう. この定式化により, 与えられたアイテムバンク・テスト構成条件から最大数の複数等質テスト出力することを本手法は数学的に保障できる. また, いくつかの条件でテスト構成を行い, 先行研究より多くのテストを構成することを確認した.

第 4 章では, 第 3 章で提案した厳密手法を乱数探索を用いて近似し, 計算量の問題を緩和する手法の提案した. 第 3 章で提案・開発した厳密手法の最大の問題は, 最大クリーク探索を行う際の空間計算量であった. 具体的には, 複数等質テスト構成のためのグラフ構造が非常に大きいため, 計算機の主記憶上に保持できない問題があった. そこで本近似手法では, グラフ全域からの探索の代わりに, ランダムに選び出した部分グラフからの探索を繰り返しにより, 空間計算量を軽減する. この近似は, グラフ全域をメモリ上に保持する代わりに部分グラフをメモリ上に保持することで, 空間計算量を軽減可能である. また, 本手法は時間漸近的に最大クリークを発見可能である. 本章では本手法の有効性を示すため, 様々な条件でテスト構成数の比較を行った. その結果本近似手法は, 厳密手法では計算量の関係で計算不能であった条件でもテスト構成可能であり, 従来手法よりも多くのテスト数を構成可能であった. ただし, テスト構成の規模に対し与える空間計算量条件が少ない場合, 従来手法よりもテスト構成数が少なくなる場合も確認された. そのような場合においても, 重複項目数を増やすことで従来手法よりも効率的にテスト構成数を増やせることを示した.

以上より本手法は, テスト間に重複を許す場合において与えられたアイテムバンクから十分な数のテストを構成することを示した. 本近似手法はリクルートキャリア社により実施される人事測定テストで実際に使用されている.

ただし, 本手法により構成されたテスト群中において, それぞれの項目の出題回数には偏りが生じえる. なぜなら, 本手法では 2 テスト間の最大重複項目数のみを制御しており, 構成テスト群全体でそれぞれの項目が何度出題されているのかは制御していない. そのため, 一部の項目のみが構成テストのほとんどすべてに出題されてしまう可能性がある. 一般にテスト運用上, このような項目使用回数の偏りは好ましくない. なぜなら, 露出の多い項目は受験者間で共有されやすく, 経年による運用において, その項目の信頼性が失われやすくなるためである. この問題は項目暴露問題と呼ばれており, 暴露について制御をおこなわない場合, 一般に構成テスト群中の項目出題頻度は Zipf の法則に従うと Wainer は報告している [42]. 今後の課題として, この項目暴露を均一にするテスト構成手法の開発に取り組んでいきたい.

また, テスト構成のためのグラフの構造についても分析を行いたい. グラフ構造をより細かく分析することで新たな特徴を発見できれば, その特徴を使い探索アルゴリズムをより効率化できる可能性がある. 具体的には、グラフの次数分布などを分析しより良い探索手法がないかを検

討していきたい.

最後に本論では, テストの目的に対してどの程度のテスト構成条件が適切なのか，項目重複を許してもよいのか, などについては議論を行っていない．この点についても今後の課題としていきたい.

参考文献

[1] ISO. *ISO/IEC 23988:2007, Information technology – A code of practice for the use of information technology (IT) in the delivery of assessments*, 2007.

[2] 植野真臣, 永岡慶三. e テスティング. 培風館, 2009.

[3] 植野真臣. e テスティング：最先端テスト技術. 電子情報通信学会誌, Vol. 92, No. 12, pp. 1017–1021, dec 2009.

[4] 植野真臣. e テスティング： 先端理論と技術. 情報システム学会誌, Vol. 26, No. 2, pp. 204–217, 2009.

[5] Frederic M. Lord. *Applications of Item Response Theory To Practical Testing Problems*. Routledge, 1st edition, July 1980.

[6] T. J. J. M. Theunissen. Binary programming and test design. *Psychometrika*, Vol. 50, No. 4, pp. 411–420, December 1985.

[7] Wim J. van der Linden and Ellen Boekkooi-Timminga. *A zero-one programming approach to Gulliksen's matched random subtest method*. Project psychometrische aspecten van item banking. Department of Education of the Universitiy of Twente, 1986.

[8] T. J. J. M. Theunissen. Some applications of optimization algorithms in test design and adaptive testing. *Applied Psychological Measurement*, Vol. 10, No. 4, pp. 381–389, 1986.

[9] Wim J. van der Linden and Ellen Boekkooi-Timminga. A maximin model for irt-based test design with practical constraints. *Psychometrika*, Vol. 54, No. 2, pp. 237–247, June 1989.

[10] Jos J. Ameda and Wim J van der Linden. Algorithms for computerized test construction

using classical item parameters. *Journal of Educational Statistics*, Vol. 14, pp. 279–290, 1989.

[11] Jos J. Ameda. Models and algorithms for the construction of achievement tests. *Ph.D. dissertation, University of Twente*, 1990.

[12] Jos J. Adema, Ellen Boekkooi-Timminga, and Wim J van der Linden. Achievement test construction using 0-1 linear programming. *European Journal of Operational Research*, Vol. 55, No. 1, pp. 103–111, 1991.

[13] Ellen Boekkooi-Timminga. Simultaneous test construction by zero-one programming. *Methodika*, Vol. 1, pp. 101–112, 1987.

[14] Frank B. Baker, Alan S. Cohen, and B. Ross Barmish. Item characteristics of tests constructed by linear programming. *Applied Psychological Measurement*, Vol. 12, No. 2, pp. 189–199, 1988.

[15] Terry A. Ackerman. An alternative methodology for creating parallel test forms using the irt information function. *Paper presented at the Annual Meeting of the National Council on Measurement in Education*, March 1989.

[16] Ellen Boekkooi-Timminga. The construction of parallel tests from irt-based item banks. *Journal of Educational Statistics*, Vol. 15, pp. 129–145, 1990. reports.

[17] Jos J. Ameda. Implementations of the branch-and-bound method for test construction problems. *Methodika*, Vol. 6, pp. 99–117, 1992.

[18] Jos J. Adema. Methods and models for the construction of weakly parallel tests. *Applied Psychological Measurement*, Vol. 16, No. 1, pp. 53–63, 1992.

[19] Len Swanson and Martha L. Stocking. A model and heuristic for solving very large item selection problems. *Applied Psychological Measurement*, Vol. 17, No. 2, pp. 151–166, 1993.

[20] Ronald D. Armstrong, Douglas H. Jones, and Charles S. Kunce. Irt test assembly using network-flow programming. *Applied Psychological Measurement*, Vol. 22, No. 3, pp. 237–247, 1998.

[21] H.L. Jeng and S.G. Shih. A comparison of pair-wise and group selections of items using

simulated annealing in automated construction of parallel tests. *Psychological Testing*, Vol. 44, No. 2, pp. 195–210, 1997.

[22] Richard M. Luecht. Computer-assisted test assembly using optimization heuristics. *Applied Psychological Measurement*, Vol. 22, No. 3, pp. 224–236, 1998.

[23] Wim J. van der Linden and Jos J. Adema. Simultaneous assembly of multiple test forms. *Journal of Educational Measurement*, Vol. 35, No. 3, pp. 185–198, September 1998.

[24] Wim J. van der Linden. *Liner Models for Optimal Test Design*. Springer, 2005.

[25] Dmitry I. Belov and Ronald D. Armstrong. Monte carlo test assembly for item pool analysis and extension. *Applied Psychological Measurement*, Vol. 29, pp. 239–261, 2005.

[26] Dmitry I. Belov and Ronald D. Armstrong. A constraint programming approach to extract the maximum number of non-overlapping test forms. *Computational Optimization and Applications*, Vol. 33, pp. 319–332, 2006.

[27] Gwo-Jen Hwang, Peng-Yeng Yin, and Shu-Heng Yeh. A tabu search approach to generating test sheets for multiple assessment criteria. *IEEE Transactions on Education*, Vol. 49, No. 1, pp. 88–97, 2006.

[28] Angela. J. Verschoor. Genetic algorithms for automated test assembly. *Ph.D. dissertation, University of Twente, Enschede*, 2007.

[29] Kejing He, Li Zheng, Shoubin Dong, Liqun Tang, Jianfeng Wu, and Chunmiao Zheng. Pgo: A parallel computing platform for global optimization based on genetic algorithm. *Computers and Geosciences*, Vol. 33, No. 3, pp. 357–366, 2007.

[30] Koun-Tem Sun, Yu-Jen Chen, Shu-Yen Tsai, and Chien-Fen Cheng. Creating irt-based parallel test forms using the genetic algorithm method. *Applied Measurement in Education*, Vol. 2, No. 21, pp. 141–161, 2008.

[31] Dmitry I. Belov. Uniform test assembly. *Psychometrika*, Vol. 73, No. 1, pp. 21–38, 2008.

[32] Pokpong Songmuang and Maomi Ueno. Bees algorithm for construction of multiple test forms in e-testing. *IEEE Transactions on Learning Technologies*, Vol. 4, pp. 209–221,

2011.

[33] 豊田秀樹. 項目反応理論. 朝倉書店, 2002.

[34] F.B. Baker and S.H. Kim. *Item Response Theory: Parameter Estimation Techniques*. Statistics Textbooks and Monographs. Marcel Dekker, 2004.

[35] Richard M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, Vol. 40, No. 4, pp. 85–103, 1972.

[36] Hiroaki Nakanishi and Etsuji Tomita. An $o(2^{0.19171n})$-time and polynomial-space algorithm for finding a maximum clique. *Information Processing Society of Japan SIG Technical Report*, Vol. 2008, No. 6, pp. 15–22, 2008.

[37] ILOG. *ILOG CPLEX User's Manual 11.0*, 2007.

[38] Francisco J. Solis and Roger J-B. Wets. Minimization by random search techniques. *Mathematics of operations research*, Vol. 6, No. 1, pp. 19–30, 1981.

[39] Qingfu Zhang, Jianyong Sun, and Edward Tsang. An evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 2, pp. 192 – 200, april 2005.

[40] Xiutang Geng, Jin Xu, Jianhua Xiao, and Linqiang Pan. A simple simulated annealing algorithm for the maximum clique problem. *Information Sciences*, Vol. 177, No. 22, pp. 5064–5071, 2007.

[41] S. Balaji, V. Swaminathan, and K. Kannan. A simple algorithm to optimize maximum independent set. *Advanced Modeling and Optimization*, Vol. 12, No. 1, pp. 107–118, 2010.

[42] Howard Wainer. Rescuing computerized testing by breaking zipf's law. *Journal of Educational and Behavioral Statistics*, Vol. 25, pp. 203–224, 2000.

[43] Takatoshi Ishii, Pokpong Songmuang, and Maomi Ueno. A method to extract the maximum number of test forms using maxclique. *The 23rd Annual Conference of the Japanese Society for Artificial Intelligence*, 2009.

[44] 石井隆稔, ソムァン・ポクポン, 植野真臣. 最大クリーク問題を用いた複数等質テスト自動構成手法. 電子情報通信学会 和文 D, Vol. J97-D, , 2014. 採録中.

[45] Takatoshi Ishii, Pokpong Songmuang, and Maomi Ueno. Maximum clique algorithm for uniform test forms. *The 16th International Conference on Artificial Intelligence in Education*, 2013.

[46] Takatoshi Ishii, Pokpong Songmuang, and Maomi Ueno. Maximum clique algorithm and its approximation for uniform test forms. *IEEE Transaction on Learning Technologies*. accepted for publication.

謝辞

本研究を進めるにあたり, 終始懇切なる御指導を賜った, 電気通信大学大学院教授の植野真臣先生と, タマサト大学講師の Pokpokng Songmuang 先生に, 心より感謝を申し上げます. 本論文の審査過程において, 数々の貴重な御助言と御指導を賜りました大須賀昭彦教授, 栗原聡教授, 田原康之准教授, 古賀久志准教授に深謝申し上げます. また, 本研究における議論・検討に当たって, ご教示とご激励を頂いた電気通信大学大学院植野真臣研究室の皆様に御礼申し上げます.

関連論文の印刷公表の方法及び時期

査読付き論文（本学位申請論文関連論文）

石井 隆稔, ソムァン・ポクポン, 植野 真臣, “最大クリーク問題を用いた複数等質テスト自動構成手法”, 電子情報通信学会 和文 D **(採録中, 受付番号 2013JDP7035,Vol.J97-D,No.2,pp.-,Feb. 2014)**

Takatohsi Ishii, Pokpong Songmuang, Maomi Ueno, “Maximum Clique Algorithm and its approximation for Uniform Test Form Assembly” IEEE Transaction on Learning Technologies. **(採録中, Manuscript ID: TLT-2013-05-0075, DOI: 10.1109/TLT.2013.2297694)**

国際会議

Takatoshi Ishii, Pokpong Songmuang , and Maomi Ueno, Maximum Clique Algorithm for Uniform Test Forms Assembly, Artificial Intelligence in Education 2013 (2013 年 7 月), Memphis Tennessee USA, Artificial Intelligence in Education Lecture Notes in Computer Science Volume 7926 2013 pp 451-462

その他（研究会等）

○石井隆稔、植野真臣、“ 最大クリーク抽出法を用いた等質テスト生成数の最大化 ”、JSISE 研究会. 論文集 (2009)

○石井隆稔, ソンムァン ポクポン, 植野真臣, ” e テスティングにおける最大クリーク抽出法

を用いた等質テスト生成数の最大化法, 人工知能学会全国大会 (第 23 回) 論文集 (2009)

○石井隆稔、植野真臣、ソンムァン ポクポン, ” ｅテスティングにおける最大クリーク抽出法を用いた等質テスト生成数の最大化法 ” 日本テスト学会第 8 回大会発表論文抄録集, (2010)

○石井 隆稔, ソンムァン ポクポン， 植野 真臣, テスト構成数を最大化する複数等質テスト自動構成手法, 日本教育工学会 第 28 回全国大会 (2012)

○石井 隆稔, ソンムァン ポクポン， 植野 真臣, 最大クリーク問題を用いた複数等質テスト自動構成手法, 教育システム情報学会 第 37 回全国大会.(2012)

○石井　隆稔, ソンムァン　ポクポン, 植野 真臣, ” e テスティングにおける複数テスト自動構成近似手法 ”, 教育情報システム学会第 38 回全国大会講演論文集 pp231-232, (2013)